Universidad de Costa Rica
Bachillerato en Computación
Seguridad de Sistemas Computacionales
CI0143 - I Semestre 2022

Seguridad de Aplicaciones Web
Fecha finalización: 25 julio

Prof. Ricardo Villalón Fonseca

1. Objetivo General

Analizar y proponer mejoras a la seguridad de aplicaciones web desarrolladas por los estudiantes del curso, a través de una dinámica de juego o competencia.

2. Objetivos Específicos

1. Diagnosticar y documentar problemas de seguridad en aplicaciones web y en la infraestructura de los servidores que las soportan.

2. Implementar ataques a las vulnerabilidades encontradas en las aplicaciones y en la infraestructura.

3. Proponer políticas y controles para mejorar el nivel de seguridad de las aplicaciones y eventuales ajustes en la configuración de los servicios de servidor que las sustentan.

3. Recomendaciones y advertencias de seguridad

Durante el curso se estudian algunos tópicos sensibles de seguridad de la información, por lo tanto, el estudiante debe velar porque las tareas asignadas sean desarrolladas en un ambiente seguro, aislado de su entorno normal de trabajo y preferiblemente desconectado de otras redes cuando sea posible.

Para este laboratorio, si usted requiere usar aplicaciones de software que sean clasificadas como software de alto riesgo debe informar con anterioridad al profesor del curso, así como aislar el uso del mismo para evitar cualquier alteración de la red en la que opera su computador, en la infraestructura de la Universidad donde está implementado el laboratorio o en cualquier otra red fuera del entorno de esta actividad académica.

4. Descripción General

El laboratorio se desarrolla a partir de un conjunto de aplicaciones web creadas previamente y dentro de una infraestructura de servidores preparada para instalar dichas aplicaciones. La dinámica es la de un juego, en donde los participantes intentan descubrir vulnerabilidades, problemas y soluciones relacionadas a la seguridad de las aplicaciones de software, así como a la infraestructura de servicios de servidor que las soportan.

El juego tiene una duración de al menos 2 semanas. Los estudiantes deben buscar vulnerabilidades y crear exploits para las aplicaciones de los otros estudiantes. En las secciones siguientes se describen los procedimientos, documentos, reglas y demás aspectos requeridos para la realización del juego. Por favor, lea las instrucciones en detalle antes de iniciar el trabajo.

5. Creación de reportes de seguridad

Durante el juego se generarán varios tipos de documentos, **los cuales deben ser subidos en formato PDF solamente (NO .doc, NO .docx, NO .odt, etc.)**, y estar asociados a los hallazgos o situaciones de seguridad encontradas en las aplicaciones o en la plataforma de servidores, a saber:

- **Anuncio de vulnerabilidad**: describe una vulnerabilidad encontrada en alguna de las aplicaciones.
- **Exploit o ataque**: es un componente de software, entrada de datos y/o secuencia de comandos, con su correspondiente documentación, que explota una vulnerabilidad o error de configuración.
- **Deficiencia o error de configuración**: describe un deficiencia o error de configuración de un servicio de servidor que soporta una aplicación web.
- **Solución**: es una propuesta de solución a una vulnerabilidad o error de configuración a su aplicación.
- **Recomendación**: es una mejora o cambio propuesto a la seguridad de una aplicación o servicio de infraestructura que la soporta.

Cada participante preparará documentos y/o software de los tipos descritos y según los hallazgos de seguridad en las aplicaciones para obtener puntaje. Adicionalmente a los reportes para competencia, los participantes pueden crear reportes de las actividades realizadas, esto con el fin de documentar los procesos de revisión de seguridad de las aplicaciones que no produzcan ninguna vulnerabilidad o exploit, y que en última instancia serían considerados para la nota del curso en esta tarea. Las fechas de las entradas en el foro del curso creado para la COMPETENCIA se considerarán como las oficiales para el recibido de documentos.

Al final de este documento se provee una guía general con los formatos para la presentación de los documentos.

6. Parámetros para evaluación de la seguridad

Uno de los objetivos importantes de la tarea es la búsqueda de vulnerabilidades o errores en la seguridad de las aplicaciones; por ello se definen los siguientes parámetros para clasificar las vulnerabilidades y/o los exploits, y que serán usados para asignar el puntaje durante la competencia.

1. **Nivel de gravedad o impacto del problema**: se refiere a la gravedad, desde la perspectiva de la aplicación, de una vulnerabilidad detectada. Para su cálculo se considerará el impacto a nivel de confidencialidad, integridad y disponibilidad, y tomando como referencia parámetros del framework CVSS para evaluación de vulnerabilidades. Hay 4 niveles para este parámetro que son *crítico*, *alto*, *medio* y *bajo*. Entre mayor sea el nivel de gravedad de una vulnerabilidad detectada, mayor será el puntaje obtenido.

2. **Complejidad de la detección/explotación**: se refiere a la complejidad del proceso realizado para detectar y/o materializar una vulnerabilidad en un exploit, entre más alto sea el nivel de complejidad del proceso, mayor será el puntaje obtenido. Para medir la complejidad también se consideran metricas definidas por el framework CVSS con respecto a complejidad, autenticación y tiempo de máquina requerido para lograr el hallazgo. Hay 4 niveles que son *muy alto*, *alto*, *medio* y *bajo*.

7. Evaluación de hallazgos de seguridad

Cada participante generará documentos con propuestas donde se presentan los hallazgos de seguridad. El proceso de generación de estos documentos debe registrarse apropiadamente a través del foro de la plataforma web del curso, es decir, las propuestas de vulnerabilidades, exploits, etc. deben enviarse al foro. Tanto el trabajo realizado que no produzca resultados concretos, como la detección de una vulnerabilidad,debe documentarse a través del foro para generar evidencia de la participación en la competencia.

Se recibirán las propuestas de forma oficial a través de la plataforma, no se recibirán propuestas enviadas por email de forma directa. Una vez enviada una propuesta, se procederá a la evaluación correspondiente. Una propuesta puede ser aceptada, modificada (en cuyo caso deberán hacerse cambios para ser aceptada) o incluso rechazada. Cuando corresponda, se emitirá la resolución por medio del foro junto con cualquier aclaración o comentario.

8. Sistema de Puntos

Los parámetros *valores de nivel de gravedad* y *complejidad del problema* se suman para dar el puntaje de cada reporte.

Para el *nivel de gravedad o impacto del problema* los puntajes son:

- Bajo: 10.
- Medio: 20.
- Alto: 35.
- Crítico: 50.

Para la *complejidad del hallazgo* los puntajes son:

- Bajo: 10.
- Medio: 25.
- Alto: 50.
- Muy alto: 100.

Un detalle importante con respecto a la liberación de vulnerabilidades y otros tipos de reportes es que una vez que se hacen públicos en el foro, no es posible reclamar puntos sobre el mismo item, los puntos son otorgados a quien realiza de primero el anuncio y luego de haber pasado el proceso de aprobación. Sin embargo, es posible presentar una versión "mejorada"de una vulnerabilidad o exploit, es decir, si para un reporte ya presentado se logra crear una nueva versión que demuestra un mayor nivel de peligrosidad o de alcance, no solamente una versión diferente de demostrar el hallazgo, entonces será valorada para su aceptación.

Las soluciones y recomendaciones tienen un valor de 15 puntos después de su aprobación.

Se han definido tres fechas, a saber, 11 de julio, 18 de julio y 25 de julio, todas ellas a las 11:55p m, como los momentos en que se harán los cortes para las entregas de analisis de seguridad.

9. Regulaciones para el uso de la infraestructura

Durante la competencia usted puede imaginar casi todo tipo de ataques a las aplicaciones y la infraestructura, pero debe respetar una serie de reglas para no dañar el trabajo del resto de participantes.

No se permite realizar ataques que afecten la operación normal de infraestructura o recursos pertenecientes a otras redes ajenas a la creada para el laboratorio, particularmente la red de la UCR. Esto deja por fuera, esencialmente, cualquier tipo de ataque de fuerza bruta o de tipo invasivo aplicado remotamente que produzca denegación de servicios o efectos similares, por cuanto su implementación podría consumir recursos requeridos para la operación normal del resto del entorno académico.

La implementación de ataques y exploits debe hacerse a modo de prueba de concepto lo cual significa que, si se logra detectar una vulnerabilidad e implementar un exploit que permita acceder a recursos no autorizados del sistema, *queda totalmente prohibido dañar de forma sensible la infraestructura de servicios o la integridad de la información almacenada en el sistema*, para que los demás participantes no vean interrumpida su trabajo en la competencia. En cambio, para demostrar la efectividad de un exploit, se puede agregar o modificar información de forma no sensible como por ejemplo cambiar una contraseña, agregar un registro a una base de datos, borrar un registro o información que sea recuperable en caso de que se requiera reactivar el servicio atacado, desplegar o mostrar información privada, etc.

En caso de detectar vulnerabilidades importantes, en donde para hacer efectivo un exploit se requiera realizar acciones que pudieran irrespetar las indicaciones aquí descritas, se debe contactar al profesor del curso para coordinar la ejecución del ataque de forma controlada.

La violación a las regulaciones indicadas conllevará una penalización a nivel individual del trabajo del curso y podría significar también una penalización en puntos en la competencia.

10. Reporte de trabajo académico

Para evaluar la parte académica de esta actividad, cada estudiante debe presentar 'al menos' 3 cortes de actividad realizada durante la competencia en las fechas indicadas anteriormente. Los reportes de seguridad para ganar puntos en la competencia cuentan como trabajo académico pero; adicionalmente, el estudiante puede generar reportes de trabajo, con un formato igual a los de competencia, sólo para efectos de documentar la actividad realizada que no produjo resultados exitosos en la identificación de problemas de seguridad. De esta forma es posible asignar un valor justo al esfuerzo realizado que no haya producido resultados exitosos. Los reportes de trabajo deben tener un nivel de profundidad apropiado, no solamente revisiones genéricas sin mayor valor académico.

Al final de la actividad se espera que cada estudiante haya profundizado en el estudio de los componentes tecnológicos estudiados durante el curso, pero que a la vez haya participado en la solución de problemas en las tecnologías, vulnerabilidades, etc.

Para las aplicaciones que contengan vulnerabilidades explotables, el grupo afectado deberá aportar una solución para cada una de ellas antes de la finalización de la competencia, dando un margen de 4 días para las correcciones después del último corte. Dicha solución aportará puntos a la competencia y se contabilizará como parte del trabajo académico. La no presentación de una solución se penaliza con un 10 % de la nota académica individual en alguno de los cortes.

Finalmente, es importante notar que la presentación tardía de los reportes de trabajo, en caso de no haber reportes de vulnerabilidades o exploits antes de cada corte, será penalizado con un 10 % de la nota obtenida en el reporte, por cada semana o fracción, contado a partir de la fecha y hora de entrega establecido.

Para referencia y ayuda en la distribución de tareas, se listan a continuación algunos tipos de ataques que pueden ser valorados para las aplicaciones:

- Ataques de buffer overflow.
- Ataques de inyección, por ejemplo: inyección SQL.
- Ataques de JavaScript y sesiones.
- Ataques al servidor web, por ejemplo: explotación de módulos instalados.
- Ataques al servidor de base de datos, por ejemplo: escalación de privilegios.
- Ataques al sistema operativo.

Además, las tecnologías usadas para la implementación de las aplicaciones son las definidas para el desarrollo de la aplicación CGI, en tarea previa a esta competencia.

11. Reportes y resultados de la competencia

Los reportes son los enviados por cada persona durante la competencia, pero adicionalmente hay que preparar dos elementos al final del curso:

1. Una política de seguridad revisada y mejorada para su aplicación.
2. Una resumen de las principales experiencias y enseñanzas de la actividad.

12. Guía para creación de reportes de seguridad

En esta sección se describen los formatos para reportar hallazgos de seguridad durante la competencia. La estructura en todos los casos es simple pues contiene una breve lista de secciones para documentar apropiadamente un evento de seguridad.

12.1. Anuncio de vulnerabilidad

Cada vulnerabilidad debe ser subida a la plataforma en un archivo con formato PDF y empacado en formato ZIP en caso de haber scripts adicionales. El nombre del archivo corresponde a un ID asignado a la vulnerabilidad y cuya nomenclatura se describe en esta sección. En el .ªsunto"del mensaje enviado al foro se debe colocar *solamente* el ID asignado al hallazgo. Los campos que se espera contenga un reporte de vulnerabilidad son:

1. **Identificador único de vulnerabilidad**: VUL-HOSTxxx-yyyyyy-ID, donde VUL indica que es una vulnerabilidad, HOSTxxx representa el último octeto de la dirección IP donde está alojada la aplicación vulnerable; las letras siguientes yyyyyy corresponden al nombre del estudiante que preparó el informe, seguido de un ID que es un número secuencial entre 01 y 99 para diferenciar unas vulnerabilidades de otras. Cada persona puede/debe llevar su propio consecutivo único.

2. **Tipo de problema**: describe el tipo de problema de seguridad desde la perspectiva de la aplicación, por ejemplo negación de servicio, violación a la política de seguridad, perdida de información, escalación de privilegios, ejecución de código, etc.

3. **Autor**: contiene el nombre del autor del hallazgo.

4. **Localización del problema**: describe el host, URL y otros detalles del componente de la aplicación donde se detectó la vulnerabilidad.

5. **Descripción del problema**: describe el problema de seguridad, puede incluir información del código fuente con el problema, explicación de cómo el programa puede ser usado de forma maliciosa para generar un problema, información del proceso realizado para detectar la vulnerabilidad y cualquier otra evidencia que sirva para sustentar el hallazgo.

6. **Nivel de gravedad propuesto para el problema**: el nivel de gravedad considerado por el equipo, debe justificarse.

7. **Descripción del Impacto**: describe el tipo de impacto que el problema puede tener en el sistema, por ejemplo, que puede producir una denegación de servicio (disponibilidad), que puede dar privilegios extra al usuario (confidencialidad), que podría permitir la modificación de registros de la base de datos (integridad), etc.

8. **Referencias**: referencias bibliográficas o sitios web donde se pueda ubicar información adicional del problema, cuando corresponda.

9. **Tiempo dedicado**: la cantidad de tiempo dedicada a este hallazgo y su documentación.

12.2. Exploit o ataque

Los exploit y ataques deben ser subidos a la plataforma en formato ZIP, incluyendo programas, scripts, datos obtenidos, etc., y la documentación solicitada en esta sección en formato PDF. Los campos que se espera contenga un exploit son:

1. **Identificador único del exploit**: utilice el mismo formato de las vulnerabilidades pero usando el prefijo EXP.
2. **Identificador de la vulnerabilidad explotada**: es el identificador único de la vulnerabilidad que es usada en este exploit.
3. **Autor**: contiene el nombre del autor.
4. **Descripción del exploit**: describe en detalle el proceso de implementación y ejecución del exploit. Esta descripción puede ir acompañada de otros documentos, como por ejemplo scripts, programas, archivos de datos, información o resultados extraídos durante el ataque, etc.
5. **Impacto**: describe el tipo de impacto que el exploit puede tener en el sistema, por ejemplo, que puede producir una denegación de servicio (disponibilidad), que puede dar privilegios extra al usuario (confidencialidad), que podría permitir la modificación de registros de la base de datos (integridad), etc.
6. **Referencias**: referencias bibliográficas o sitios web donde se pueda ubicar material de apoyo al exploit, cuando corresponda.
7. **Tiempo dedicado**: la cantidad de tiempo dedicada a este exploit y su documentación.

12.3. Errores o deficiencias de configuración

Las errores o deficiencias de configuración deben ser subidos a la plataforma en un archivo PDF o ZIP (en caso de haber scripts) y conteniendo la documentación solicitada en esta sección. Los campos que se espera contenga un reporte de error o de configuración son:

1. **Identificador único del error**:utilice el mismo formato de las vulnerabilidades pero usando el prefijo CFG.
2. **Autor**: contiene el nombre del autor del hallazgo.
3. **Descripción del error**: describe en detalle el error de configuración encontrado y que podría ser utilizado para abusar de los recursos de una aplicación o del sistema. Esta descripción puede ir acompañada de otros documentos, como por ejemplo scripts, programas, archivos de datos, información y resultados obtenidos en el hallazgo.
4. **Impacto**: describe el tipo de impacto que el problema puede tener en el sistema, por ejemplo, que puede producir una denegación de servicio (disponibilidad), que puede dar privilegios extra al usuario (confidencialidad), que podría permitir la modificación de registros de la base de datos (integridad), etc.
5. **Referencias**: referencias bibliográficas o sitios web donde se pueda ubicar material de apoyo, cuando corresponda.

6. **Tiempo dedicado**: la cantidad de tiempo dedicada a este hallazgo y su documentación.

12.4. Solución

Las soluciones a exploits y errores deben ser subidas a la plataforma en un archivo PDF o ZIP (en caso de haber scripts o programas) y conteniendo la documentación solicitada en esta sección.

1. **Identificador único de solución**: utilice el mismo formato de las vulnerabilidades pero usando el prefijo SOL.
2. **Identificador de la vulnerabilidad o error**: es el identificador único de la vulnerabilidad o error referenciada en esta solución.
3. **Autor**: contiene el nombre del autor de la solución.
4. **Descripción de la solución**: describe en detalle la solución definida. Esta descripción puede ir acompañada de otros documentos, como por ejemplo scripts, programas, archivos de datos, etc.
5. **Referencias**: referencias bibliográficas o sitios web donde se pueda ubicar material de apoyo, cuando corresponda.
6. **Tiempo dedicado**: lla cantidad de tiempo dedicada a esta solución y su documentación.

12.5. Recomendación

Las recomendaciones deben ser subidas a la plataforma en un archivo PDF y conteniendo la documentación solicitada en esta sección.

1. **Identificador único de sugerencia**: utilice el mismo formato de las vulnerabilidades pero usando el prefijo SUG.
2. **Identificador de la vulnerabilidad o error**: es el identificador único de la vulnerabilidad o error referenciada en esta solución, si la hay.
3. **Autor**: contiene el nombre del autor de la recomendación.
4. **Recomendación**: describe la recomendación del caso. Esta descripción puede ir acompañada de otros documentos, como por ejemplo scripts, programas, archivos de datos, etc.
5. **Referencias**: referencias bibliográficas o sitios web donde se pueda ubicar material de apoyo, cuando corresponda.
6. **Tiempo dedicado**: la cantidad de tiempo dedicada a esta recomendación y su documentación.