Universidad Nacional del Litoral

Proyecto Final de Carrera

Diseño de un sistema de detección de anomalías en redes de computadoras.

Pineda Leandro

dirigido por

Santa Fe
22 de agosto de 2016

Resumen

En la actualidad las redes informáticas están presentes en nuestro día a día, así como también en las operaciones diarias de prácticamente cualquier organización. Debido a la demanda de contenido multimedia, la telefonía celular y el surgimiento del *BigData*, el tamaño y el volumen de datos que manejan las empresas proveedoras de este tipo de servicios incrementa año tras año. Por esto, las herramientas que facilitan la tarea de administración y diagnóstico de redes son de gran utilidad para ofrecer un servicios de mejor calidad, y aprovechar la utilización de recursos existentes. El documento describe un proyecto para el desarrollo de un sistema de monitoreo en tiempo real del tráfico de red, que puede ser puesto en funcionamiento en la infraestructura existente de cualquier organización con un bajo impacto y a un bajo costo.

Palabras claves *tiempo real*, *redes informáticas*, *Data Streaming*, *monitoreo*, *visualización*

Justificación

La cantidad de servicios que ofrecen organismos públicos y privados a través de Internet aumenta constantemente, y en consecuencia también lo hace la complejidad de los sistemas necesarios para proveerlos.

La integridad de estos sistemas puede verse comprometida fundamentalmente por dos factores: las fallas de los componentes de la infraestructura de red[1] que afectan una parte o la totalidad de las funcionalidades del sistema, y los servicios ofrecidos a usuarios externos que pueden ser objeto de numerosos tipos de ataques[2][3]. Hacer frente a estos últimos es de extrema importancia dado que su fin puede ser no solo afectar la calidad de los servicios ofrecidos, sino que también pueden comprometer información sensible de la organización o de los usuarios. Además de ataques externos, los sistemas pueden ser vulnerados por usuarios malintensionados con acceso físico a la red interna de la organización, o por actores inadvertidos[4][5][6][7].

El conjunto de herramientas clásicas para implementar políticas de seguridad como firewall, lista de control de acceso, credenciales de usuario, proxy de red entre otros, resultaba adecuado 10 años atras, pero en la actualidad estas herramientas están siendo cada vez menos efectivas en la tarea de bloquear ataques dirigidos y malware avanzados. Esto hace que cada vez sea más común la utilización de otras herramientas que monitorean continuamente eventos en busca de patrones que puedan indicar comportamientos anómalos provocados por ataques o por fallas en los dispositivos de red. Los sistemas de detección/prevención de intrusiones (IDS o IPS por las siglas en inglés *Intrusion Detection/Prevention System*) pueden realizar esta tarea utilizando una de estas técnicas de detección: detección de uso indebido y detección de anomalías[8]. En el primer caso, el sistema determina la ocurrencia de ataques comparando la actividad de la red y de los subsistemas que la componen contra un conjunto de patrones de ataques conocidos. En el segundo caso, el sistema utiliza un modelo de comportamiento normal para evaluar la actividad de la red en busca de anomalías, las cuales pueden ser producto de fallas o potenciales ataques. Como alternativa a las soluciones de hardware dedicado o sistemas privativos que ofrecen algunas empresas, la comunidad de software libre desarrolla y mantiene la aplicación Snort[1], que es un IDS basado en detección de uso indebido. Si bien esta herramienta es efectiva detectando ataques conocidos, es dependiente de la definición de un conjunto efectivo de reglas por parte del administrador (las cuales son difíciles de mantener) y de una base de datos actualizada con los patrones conocidos de ataques o *malware*. Otra desventaja de esta técnica de detección es la imposibilidad de detectar ataques *zero-day*[2][8]. Otra herramienta que suele utilizarse en conjunto con Snort es Bro[3], que es una plataforma para análisis de tráfico de red. Sin embargo, Bro también se basa en la definición de reglas de monitoreo y utiliza detección de uso indebido.

Con el fin de disponer de otra herramienta para la identificación de comportamientos sospechosos, el objetivo de este proyecto es el de diseñar un software que modele de forma automática el comportamiento normal de la red utilizando las técnicas del estado del arte, para identificar y alertar comportamiento anómalo en la red de datos de una organización. Así, el modelo de comportamiento normal es calculado automaticamente por el software, brindando la posibilidad de detectar efectivamente cualquier cambio poco usual en los sistemas. Para esto, se hará uso de la información provista por los dispositivos de infraestructura de red como routers y logs de servidores y de los subsistemas que interactuan dentro de la red.

[1] https://www.snort.org/

[2] ataques que explótan vulnerabilidades que aún no han sido conocidas o que no están contempladas en al conjunto definido de reglas

[3] https://www.bro.org/

Objetivos

Generales

Diseñar un sistema que permita monitorear en tiempo real parámetros de performance del tráfico que atraviesa una interfaz de red.

Específicos

- Identificar un conjunto de tecnologías y herramientas adecuadas para el diseño del sistema.
- Diseñar un sistema de monitoreo de redes que funcione en *commodity hardware*.
- Construir una herramienta de software que registre los eventos ocurridos en la red (capa de transporte).
- Elaborar un documento que describa la arquitectura del sistema.

Alcances

Funcionales

- El sistema permitirá visualizar en tiempo real los parámetros de performance de la red. Se podrá monitorear el tráfico a nivel de *host* (capa de red) y a nivel de servicios (capa de transporte)
- Los informes mostrarán información de la Capa de Transporte y la Capa de Red[4]. [9][10][11]
- El sistema proveerá al usuario una interfaz para la visualización de los informes. Estos contaran con los siguientes parámetros:
  - Utilización de la red
  - Paquetes por segundo transmitidos
  - Cantidad de peticiones de conexión
  - Cantidad de conexiones realizadas
  - Cantidad de conexiones rechazadas y expiradas
  - Cantidad de conexiones activas
  - Tráfico entrante y saliente por host
  - Tráfico entrante y saliente por proceso
- El sistema almacenará información de las conexiones establecidas. Lo hará en archivos de texto plano y en formato semi-estructurado.
- El sistema no identificará patrones o generará alarmas.

No Funcionales

- El sistema será escalable. Las tecnologías seleccionadas permiten configurar un cluster de nodos de procesamiento en caso que el volumen de datos a procesar aumente.
- Se proveerá un manual de instalación y configuración.

[4] Modelo TCP/IP

Exclusiones

El proyecto no contempla la instalación del sistema en un ámbito de producción. El sistema a desarrollar solo analizará el tráfico que atraviesa una única interfaz de red, y no contempla el caso de conexiones múltiples con balance de carga, ni la utilización de múltiples nodos configurados en modo cluster. No se implementará ningún mecanismo de detección ataques o de patrones de comportamiento sospechoso.

Supuestos

Los datos necesarios para realizar pruebas pueden ser generados arbitrariamente. El término *tiempo real* refiere al denominado *tiempo real blando*, es decir, no es necesario asegurar la ejecución de ciertas tareas o mostrar informes en el mismo instante en el que se generan los datos. La generación de los informes tendrá entonces demoras de cientos de milisegundos.

Criterios de Aceptación

Se considera que el proyecto está aceptado cuando cumple en un 90 % los requisitos funcionales, con un nivel mínimo de aceptación del 75 %. Los prototipos deben tener implementadas todas las funcionalidades planificadas.

Metodología

La metodología de desarrollo del proyecto será incremental e iterativa. Incremental porque varias componentes del sistema se desarrollarán en momentos diferentes y serán integradas cuando sean completadas. Iterativa pues se invertirán esfuerzos en revisar constantemente partes del sistema, tanto para mejorar la calidad externa como interna del software[12].

Con esta metodología se puede dividir el trabajo en incrementos que son revisados constantemente a lo largo de la ejecución del proyecto. Luego de una investigación preliminar acerca de las tecnologías disponibles y sus características, correspondiente a la primera fase, se tiene el punto de partida para comenzar el desarrollo. Dado que es necesario que el sistema cumpla con los requisitos de *tiempo real*, el incremento donde se implementan las funcionalidades esenciales de captura de paquetes es revisado constantemente para mejorar su calidad mientras se desarrollan los incrementos de informes y visualización (ver Plan de Tareas). Una vez todos los incrementos son terminados, serán integrados y se llevarán a cabo las pruebas de integración correspondientes.

Plan de Tareas

El proyecto se divide en 4 incrementos. A continuación se da una breve descripción de los mismos:

Incremento 1: Diseño La primer parte del proyecto consiste en determinar que conjunto de tecnologías serán utilizadas, y elaborar una descripción a alto nivel de las diferentes componentes del sistema y cómo se relacionan.

Incremento 2: *Core* o Núcleo En esta etapa se implementa la funcionalidad que permite capturar la información de entrada, visualizarla en texto plano y prepararla para procesamientos posteriores. También se implementa la funcionalidades de registro de eventos en la capa de transporte.

Incremento 3: Filtrado Se implementa el módulo que procesa el tráfico de la red y se implementan las funcionalidades de filtrado junto con los filtros predeterminados.

Incremento 4: Visualización En este incremento se integran las funcionalidades de captura y filtrado. Además se implementa la interfaz de usuario en la plataforma determinada en la etapa de diseño.

Al finalizar la primera iteración de cada incremento se obtiene una herramienta de software con las funcionalidades descriptas y calidad de producto final, con excepción de la etapa de diseño donde se obtendrá un informe en soporte escrito o digital.

Entregable	Fecha de entrega
Informe de avance 1	16/09/2016
Informe de avance 2	04/11/2016
Informe de avance 3	10/02/2016
Informe de avance 4	17/03/2017

Cuadro 1: Fechas de entrega de informes de avance.

Plan de Tareas

La duración total del proyecto es de 520 horas, con una dedicación de 20 horas semanales. A continuación se define el plan de tareas.

1. **Diseño** (84hs)
   1.1. Estudio comparativo de las tecnologías disponibles. (24hs)
   1.2. Diseño conceptual del sistema. (40hs)
   1.3. Diseño de interfaz de usuario y filtros predeterminados. (20hs)
2. ***Core* o Núcleo** (120hs)
   2.1. Instalación y configuración de servidor de aplicación. (32hs)
   2.2. Implementación de funcionalidad de captura de tráfico de red. (52hs)

2.3. Implementación de funcionalidad de registro de eventos. (24hs)

2.4. Documento de instalación. (12hs)

3. **Filtrado y visualización** (176hs)

3.1. Implementación de funcionalidad de filtrado en capa de red. (52hs)

1) Módulo de filtrado por host. (20hs)
2) Módulo de obtención de métricas. (32hs)

3.2. Implementación de funcionalidad de filtrado en capa de transporte. (60hs)

1) Módulo de filtrado por servicio. (20hs)
2) Módulo de obtención de métricas. (40hs)

3.3. Implementación de filtros predeterminados. (24hs)

3.4. Implementación de interfaz de usuario. (40hs)

4. **Integración** (140hs)

4.1. Integración de funcionalidades e interfaz de usuario. (60hs)

4.2. Pruebas de integración. (40hs)

4.3. Elaboración de informe. (40hs)

Informes de avance

Se presentarán 4 informes de avance en las fechas de finalización de cada etapa, detalladas en el Cuadro 2. A continuación se detalla que información será incluida en cada informe:

Informe de avance 1 Contendrá los resultados obtenidos en los estudios comparativos de las tecnologías y justificará la elección de las mismas. Además se incluirá una descripción general de la arquitectura del sistema.

Informe de avance 2 Contendrá información sobre el desempeño del núcleo del sistema. Se proveerá la guía de instalación y configuración de la plataforma. Además tendrá información sobre cambios realizados en los entregables anteriores.

Informe de avance 3 Contendrá una descripción de los mecanismos de filtrado utilizados y los resultados obtenidos. Además tendrá información sobre cambios realizados en los entregables anteriores.

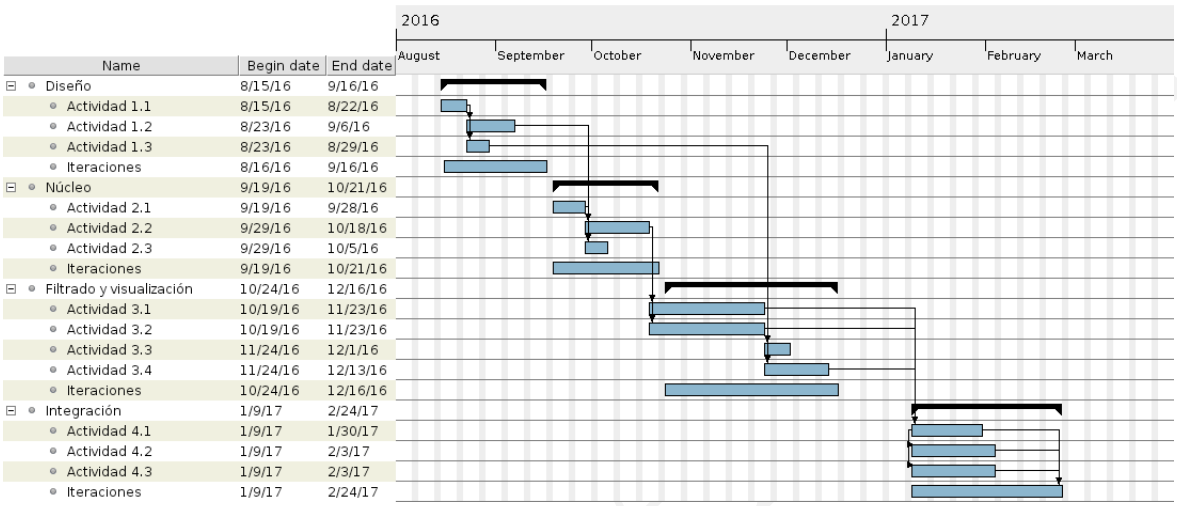
Etapa	Inicio	Finalización	Duración
Diseño	15/08/2016	23/09/2016	5 semanas
Core o Núcleo	26/09/2016	11/11/2016	6 semanas
Filtrado y visualización	14/11/2016	03/02/2017	11 semanas
Integración	06/02/2017	31/03/2017	7 semanas

Cuadro 2: Fechas estimativas de inicio y fin de actividades.

Informe de avance 4 Contendrá una información sobre los avances en el desarrollo de la interfaz de usuario. Se describirán las pruebas de integración del sistema. Además tendrá información sobre cambios realizados en los entregables anteriores.

Diagrama de Gantt

A continuación se muestra el diagrama de Gantt del proyecto.



Name	Begin date	End date
Diseño	8/15/16	9/16/16
Actividad 1.1	8/15/16	8/22/16
Actividad 1.2	8/23/16	9/6/16
Actividad 1.3	8/23/16	8/29/16
Iteraciones	8/16/16	9/16/16
Núcleo	9/19/16	10/21/16
Actividad 2.1	9/19/16	9/28/16
Actividad 2.2	9/29/16	10/18/16
Actividad 2.3	9/29/16	10/5/16
Iteraciones	9/19/16	10/21/16
Filtrado y visualización	10/24/16	12/16/16
Actividad 3.1	10/19/16	11/23/16
Actividad 3.2	10/19/16	11/23/16
Actividad 3.3	11/24/16	12/1/16
Actividad 3.4	11/24/16	12/13/16
Iteraciones	10/24/16	12/16/16
Integración	1/9/17	2/24/17
Actividad 4.1	1/9/17	1/30/17
Actividad 4.2	1/9/17	2/3/17
Actividad 4.3	1/9/17	2/3/17
Iteraciones	1/9/17	2/24/17

Figura 1: Diagrama de Gantt del proyecto.

Riesgos

En esta sección se enumeran los riesgos identificados, indicadores y estrategia a adoptar según corresponda. Para realizar el análisis cualitativo de los riesgos se asigna una probabilidad de ocurrencia y un impacto a cada riesgo. Luego se priorizan según su *severidad*. Los riesgos identificados son:

1. **No se pueden satisfacer las restricciones de performance:** la característica principal del sistema es la posibilidad de procesar tráfico en tiempo real. Puede darse un escenario donde las tecnologías disponibles no permitan alcanzar este requisito.

   Indicador: se observan demoras o un progreso lento en las etapas de Núcleo y/o Integración.
   Probabilidad de ocurrencia: Media.
   Impacto: Muy alto.
   Estrategia a adoptar: Mitigar (probabilidad). Se revisan continuamente los entregables de la etapa de Diseño con el fin de mejorar su calidad.

2. **No se dispone del hardware necesario:** el servidor de aplicaciones necesario para el desarrollo de la aplicación no se encuentra disponible

	1	2	3	4	5
1					
2					
3					
4					
5					

Cuadro 3: Probabilidad/impacto.

Severidad	Estrategia
menor que 4	Aceptar
5 a 15	Mitigar
16 a 25	Evitar

Cuadro 4: Estrategia según severidad.

Indicador: la etapa de Núcleo no puede comenzar.
Probabilidad de ocurrencia: Media.
Impacto: Alto.
Estrategia a adoptar: Mitigar (impacto). Se utilizará la infraestructura provista por la facultad.

3. **Los módulos del sistema no pueden ser integrados correctamente:** aunque cada uno de los módulos cumpla con los requisitos funcionales del sistema, puede ocurrir que el rendimiento se vea degradado producto de los retardos que puede introducir las interfaces de comunicación entre los módulos.

   Indicador: se observan demoras o un progreso lento en la etapa de Integración.
   Probabilidad de ocurrencia: Baja.
   Impacto: Muy alto.
   Estrategia a adoptar: Mitigar (probabilidad). Se revisan continuamente los entregables de la etapa de Integración con el fin de mejorar su calidad.

4. **Las tecnologías seleccionadas no pueden ser integradas:** se utilizarán un conjunto de tecnologías que deben coexistir para realizar el procesamiento de los datos. Es posible que se encuentren incompatibilidades entre alguna de ellas y no sea posible utilizarlas de forma conjunta.

   Indicador: se observan demoras o un progreso lento en la etapa de Diseño.
   Probabilidad de ocurrencia: Baja.
   Impacto: Alto.
   Estrategia a adoptar: Mitigar (impacto).
   Se realiza un estudio comparativo de varias tecnologías para disponer de alternativas a las determinadas inicialmente.

5. **El progreso del proyecto no es monitoreado adecuadamente:** es necesario elaborar una cierta cantidad de informes de avance. Puede ocurrir que debido a un monitoreo pobre del avance del proyecto estos documentos estén incompletos para la fecha de entrega.

   Probabilidad de ocurrencia: Media.
   Impacto: Bajo.
   Estrategia a adoptar: Mitigar (ocurrencia).
   Los cambios y avances realizados serán registrados mediante un sistema de control de versiones, agregando una breve descripción del trabajo realizado. Estas descripciones serán luego utilizadas para la elaboración de los informes de avance.

Análisis Riesgos

En esta sección se muestra el análisis realizado de los riesgos del proyecto, luego se definen las estrategias a adoptar y los riesgos ordenados por importancia según su severidad.

Riesgo	Imp.	% ocur.	Sev.
No se pueden satisfacer las restricciones de performance.	5	3	15
No se dispone del hardware necesario.	4	3	12
Los módulos del sistema no pueden ser integrados correctamente.	5	2	10
Las tecnologías seleccionadas no pueden ser integradas.	4	2	8
El progreso del proyecto no es monitoreado adecuadamente.	2	3	6

Cuadro 5: Lista de riesgos ordenados por severidad.

Presupuesto

A continuación se detalla el presupuesto necesario para el desarrollo del proyecto. El servidor de desarrollo cuenta con hardware de red específico y será utilizado para ejecutar las pruebas del sistema.

Bienes de capital			
Estación de trabajo (costo de amortización)			$2691.
Procesador AMD FX6			
Memoria 8GB DDR3			
Monitores (configuración dual)			
Servidor de desarrollo			$18000.
Procesador Intel i5 o similar			
Memoria 16GB DDR3			
Interfaz de red para servidor (2 puertos)			
Impresora (costo de amortización)			$318
Servicios			
Servicio de Internet			$4000
Insumos			
Resma de Hojas			$80
Toner			$200
Artículos de librería			$150
Recursos humanos	**Costo por hora**	**Horas**	
Director de proyecto	$150	40	$6000
Diseñador	$110	84	$9420
Desarrollador	$100	364	$36400
Tester	$85	40	$3400
Costo total			$**80659**.

Bibliografía

[1] P. Gill, N. Jain y N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications", *SIGCOMM Comput. Commun. Rev.*, vol. 41, n.º 4, págs. 350-361, ago. de 2011, ISSN: 0146-4833. DOI: 10.1145/2043164.2018477. dirección: http://doi.acm.org/10.1145/2043164.2018477.

[2] S. Karumanchi y A. C. Squicciarini, "In the wild: A large scale study of web services vulnerabilities", en *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ép. SAC '14, Gyeongju, Republic of Korea: ACM, 2014, págs. 1239-1246, ISBN: 978-1-4503-2469-4. DOI: 10.1145/2554850.2555010. dirección: http://doi.acm.org/10.1145/2554850.2555010.

[3] P. Mutchler, A. Doupé, J. Mitchell, C. Kruegel y G. Vigna, "A large-scale study of mobile web app security", en *Proceedings of the Mobile Security Technologies Workshop (MoST)*, mayo de 2015.

[4] "Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists", *Applied Ergonomic*,

[5] S. Kraemer, P. Carayon y J. Clem, "Human and organizational factors in computer and information security: Pathways to vulnerabilities", *Computers & Security*, vol. 28, n.º 7, págs. 509-520, 2009, ISSN: 0167-4048. DOI: http://dx.doi.org/10.1016/j.cose.2009.04.006". dirección: http://www.sciencedirect.com/science/article/pii/S0167404809000467.

[6] D. Liginlal, I. Sim y L. Khansa, "How significant is human error as a cause of privacy breaches? an empirical study and a framework for error management", *Computers & Security*, vol. 28, n.º 3–4, págs. 215 -228, 2009, ISSN: 0167-4048. DOI: http://dx.doi.org/10.1016/j.cose.2008.11.003. dirección: http://www.sciencedirect.com/science/article/pii/S0167404808001181.

[7] M. Ahmed, L. Sharif, M. Kabir y M. Al-maimani, *Human errors in information security*, 2012.

[8] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer y B. D. Payne, "Evaluating computer intrusion detection systems: A survey of common practices", *ACM Comput. Surv.*, vol. 48, n.º 1, 12:1-12:41, sep. de 2015, ISSN: 0360-0300. DOI: 10.1145/2808691. dirección: http://doi.acm.org/10.1145/2808691.

[9] J. Postel, *Internet protocol*, RFC 791 (INTERNET STANDARD), Updated by RFCs 1349, 2474, 6864, Internet Engineering Task Force, sep. de 1981. dirección: http://www.ietf.org/rfc/rfc791.txt.

[10] ——, *Transmission control protocol*, RFC 793 (INTERNET STANDARD), Updated by RFCs 1122, 3168, 6093, 6528, Internet Engineering Task Force, sep. de 1981. dirección: http://www.ietf.org/rfc/rfc793.txt.

[11] T. Socolofsky y C. Kale, *Tcp/ip tutorial*, RFC 1180 (Informational), Internet Engineering Task Force, ene. de 1991. dirección: `http://www.ietf.org/rfc/rfc1180.txt`.

[12] ISO/IEC, *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.

[13] M. Xie, S. Han, B. Tian y S. Parvin, "Anomaly detection in wireless sensor networks: A survey", *J. Netw. Comput. Appl.*, vol. 34, n.º 4, págs. 1302-1325, jul. de 2011, ISSN: 1084-8045. DOI: `10.1016/j.jnca.2011.03.004`. dirección: `http://dx.doi.org/10.1016/j.jnca.2011.03.004`.

[14] A. Tanenbaum, *Computer Networks*, 4th. Prentice Hall Professional Technical Reference, 2002, ISBN: 0130661023.

[15] W. Stallings, *Data and Computer Communications (5th Ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997, ISBN: 0-02-415425-3.

[16] Q. Anderson, *Storm Real-Time Processing Cookbook*. Packt Publishing, 2013, ISBN: 1782164421, 9781782164425.

[17] P. Hunt, M. Konar, F. P. Junqueira y B. Reed, "Zookeeper: Wait-free coordination for internet-scale systems", en *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, ép. USENIXATC'10, Boston, MA: USENIX Association, 2010, págs. 11-11.

[18] F. Junqueira y B. Reed, *ZooKeeper: Distributed Process Coordination*. O'Reilly Media, 2013, ISBN: 9781449361280.

[19] J. Leibiusky, G. Eisbruch y D. Simonassi, *Getting Started with Storm*. O'Reilly Media, Inc., 2012, ISBN: 1449324010, 9781449324018.