UNIVERSIDAD TECNOLÓGICA NACIONAL

Faculta regional Villa María

Ingeniería en Sistemas de Información

PROYECTO DE INVESTIGACIÓN - ¿Nombre?

Cecilia Belén Borello

Docentes a cargo: *Dra. Verónica S. Bogado*

Dr. Jorge A. Palombarini

Villa María, Córdoba, Argentina - 2019

INTRODUCCIÓN

"*La seguridad es una medida de la capacidad del sistema para proteger los datos y la información de accesos no autorizados. Sin dejar de proporcionar acceso a personas y sistemas que si lo están.*" (Software Architecture in Practice, Third Edition, 2013). A partir de la anterior definición podemos destacar la importancia de la evaluación de la seguridad de un software. Esta nos permitirá saber cuáles son las probabilidades de que nuestra información o nuestros servicios sean denegados, imposibilitándonos la correcta ejecución de nuestras actividades.

Es por esta razón que considero sumamente importante la posibilidad de simulación de ataques a un software, debido a que esto permitirá evaluar las consecuencias de un ataque real. También posibilitara tener un conocimiento previo de cómo reaccionar ante estas situaciones, que acciones deben llevarse a cabo en caso de ocurrir. Además, proporciona métodos para determinar cómo puede lograrse una prevención de estos ataques, dándonos la posibilidad de que estos no ocurran, ya que se poseerán las técnicas adecuadas para evitarlos.

En caso de que los ataques fueran exitosos, tener la posibilidad de simularlos previamente, podrá darnos herramientas para determinar cuáles son las acciones más convenientes para detenerlo, o si es mejor solo dejar que estos concluyan. Esto podría suceder cuando los costos de prevención son más elevados que los de reparación de los posibles daños efectuados por el malware.

Los resultados obtenidos de las simulaciones también nos proporcionaran información de que tan segura se encuentra nuestra información y nuestros servicios, y a partir de ella, podremos determinar si es necesario recurrir a mejores métodos de respaldo o mejores técnicas de protección de los mismos.

CARACTERÍSTICAS DE LA SEGURIDAD

La seguridad de los sistemas de información posee tres características básicas: confidencialidad, integridad y disponibilidad.

La **confidencialidad** hace referencia a la protección de la información y los servicio los frente al acceso de individuos no autorizado. Por su parte, la **integridad** estipula que ni la información ni los servicios pueden ser expuestos a manipulación no autorizada. Por último, al hablar de **disponibilidad**, el objetivo es solo permitir el uso legítimo del sistema.

Existen otras características que se utilizan para apoyar a las mencionadas anteriormente, estas son: autenticación, no rechazo y autorización.

La **autenticación** se encarga de verificar las identidades de los participantes en una transacción y comprueba si son realmente quienes dicen ser. El **no rechazo** debe garantizar que el remitente de un mensaje no puede negar haber enviado el mensaje, y que el destinatario no pueda negar haberlo recibido. En cuanto a la **autorización**, otorga a los diferentes usuarios legítimos los privilegios para realizar una tarea.

Para lograr la seguridad de un sistema de información debemos lograr detectar, reaccionar y recuperarnos de los ataques ocurrentes. Los objetos que necesitan protección son datos en reposo, datos en tránsito y procesos computacionales.

FORMALISMO DEVS

"*El formalismo DEVS fue concebido como una herramienta general de modelización y simulación de Sistemas de Eventos Discretos.*

DEVS permite representar cualquier sistema que experimente un número finito de cambios (eventos) en cualquier intervalo de tiempo. De esta forma, podremos ver que DEVS es en realidad un caso particular de la representación general de sistemas dinámicos, en la cual las trayectorias de entrada estarán restringidas a segmentos de eventos y la función de transición tendrá una forma especial que limitará las trayectorias de salida para que tengan idéntica naturaleza." (Introducción a la Modelización y Simulación de Sistemas de Eventos Discretos con el Formalismo DEVS, Ernesto Kofman).

Los anteriormente mencionados "*eventos*" representan un cambio instantáneo en alguna parte de un sistema. Estos pueden caracterizarse por un valor y un instante en el que ocurre. El valor puede ser un número, un vector, una palabra o, en general, un elemento cualquiera de un conjunto determinado.

Un modelo DEVS recibirá determinados eventos de entrada y, según las trayectorias especificados y sus condiciones iniciales, generará los correspondientes eventos de salida.

Un modelo DEVS atómico queda definido por la siguiente estructura:

$$\mathbf{M = (X, Y, S, \delta int, \delta ext, \lambda, ta)}$$

donde:

⟹ **X,** es el conjunto de valores de eventos de entrada.
⟹ **Y,** es el conjunto de valores de eventos de salida.
⟹ **S,** es el conjunto de valores de estado.
⟹ **δint,** tras ta(s1) unidades de tiempo el sistema realiza una transición interna yendo a un nuevo estado s2. El nuevo estado se calcula como **s2 = δint(s1)**. La función **(δint: S → S)** se llama **Función de Transición Interna** (Internal Transition Function).
⟹ **δext,** cuando llega un evento de entrada el estado cambia instantáneamente; el nuevo estado se calcula como **s4 = δext (s3, e, x1)** (notar que ta(s3) > e). La función **(δext: $S \times R_0^+ \times X \rightarrow S$)** se llama **Función de Transición Externa** (External Transition Function). Durante una transición externa no se produce ningún evento de salida.
⟹ **λ**, cuando el estado va de s1 a s2 se produce también un evento de salida con **valor y1 = λ(s1)**. La función **(λ: S → Y)** se llama **Función de Salida** (Output Function).
⟹ **ta,** cada posible estado s (s ∈ S) tiene asociado un avance de tiempo calculado por la **Función de Avance de Tiempo** (Time Advance Function) **(ta(s): $S \rightarrow R_0^+$)**.

Con el fin de simular sistemas más complejos, a partir de estos modelos atómicos, se pueden construir modelos acoplados. Una de las propiedades fundamentales del acoplamiento modular DEVS es la **clausura**. El cumplimiento de esta propiedad garantiza que el acoplamiento de modelos DEVS define un nuevo modelo DEVS equivalente. Esto implica que un modelo DEVS acoplado puede utilizarse a su vez dentro de un modelo más complejo de acoplamiento, dando paso a lo que se denomina acoplamiento jerárquico.

Un modelo acoplado clásico con puertos se define mediante la siguiente tupla de elementos:

$$\mathbf{N = (X; Y; D; \{M_d | \ d \in D\}, EIC, EOC, IC, Select)}$$

donde cada elemento se define de la siguiente forma:

- ⇨ **X**, conjunto de puertos y valores de entrada: $\mathbf{X = f\ (p; v)\ |p \in IPorts; v \in X_p\}}$ donde IPorts es el conjunto de puertos de entrada y Xp es el conjunto de valores para el puerto p.
- ⇨ **Y**, conjunto de puertos y valores de salida: $\mathbf{Y = \{(p; v)\ |p \in OPorts; v \in Yp\}}$ donde OPorts es el conjunto de puertos de salida e Yp es el conjunto de valores para el puerto p.
- ⇨ **D** es el conjunto de referencias a los componentes del modelo acoplado, tal que: $\forall \mathbf{d \in D}$, M_d es el modelo DEVS del componente d.
- ⇨ **EIC**, acoplamiento de entrada externo: $\mathbf{EIC \subseteq \{((N; ip_N), (d; ip_d))\ |\ ip_N \in IPorts; d \in D; ip_d \in IPorts_d\}}$
- ⇨ **EOC**, acoplamiento de salida externo: $\mathbf{EOC \subseteq \{((d; op_d); (N; op_N))\ |op_N \in OPorts; d \in D; op_d \in OPorts_d\}}$
- ⇨ **IC**, acoplamiento Interno: $\mathbf{IC \subseteq \{((a; op_a); (b; ip_b))\ |a; b \in D; op_a \in OPorts_a; ip_b \in IPorts_b\}}$ donde $((a; op_a); (b; ip_b)) \in IC \Longrightarrow a \neq b$
- ⇨ **Select**: $\mathbf{2^D \subseteq \{\ \} \rightarrow D}$ Función de desempate, la cual evita ejecución de eventos en paralelo, serializando la ejecución de los cálculos de los componentes inminentes.

DATOS PARA EL MODELADO

Para el desarrollo de la simulación son necesarios los datos respecto a que tan probable es que un sistema sea infectado por un malware; persiguiendo el objetivo de que el resultado sea lo más realista posible.

Los datos utilizados para el desarrollo del modelo matemático se basaren en el estudio "*Probability Analysis of Cyber Attack Paths*" (Análisis de probabilidad de rutas de ataque cibernético) realizado en 2013 en la Facultad de Ingeniería y Ciencias Matemáticas, de la universidad de Londres, Reino Unido.

Dicha investigación identifico cinco rutas de ataque factibles y que se asocian con infecciones reales de malware. Estas son: ataques pagados, computadoras domesticas; dispositivos móviles, suplantación de identidad (phishing) y ataques directos.

Los resultados que se consideraron relevantes para la investigación y que formaran parte del modelo son:

- ⇝ ***"Mula" pagada, ya sea sobornada o coaccionada*:** se estipula que la probabilidad de sobornar o coaccionar a un empleado para efectuar la infección es del 12%; posteriormente, la probabilidad de que se concrete el ataque es de un 80%. Por lo que la probabilidad de que se realice exitosamente un ataque por esta ruta es de **9,6%**
- ⇝ ***Ataque directo en una PC doméstica:*** entendemos por PC doméstica, a aquellas computadoras alojadas en las instalaciones de la organización. Se supone que un atacante requiere el control total de una PC doméstica comprometida para controlar de forma remota la propagación de virus. Se evaluó la probabilidad de ataque considerando las aplicaciones de seguridad y la tecnología cambiante. En base a lo anteriormente mencionado, se estipulo que la probabilidad de un ataque exitoso es del **63%**.

- ***Sitios web para compartir software (descarga de torrents):*** Se considera la fuente de infección más peligrosa. La probabilidad de infectar una computadora doméstica con dicho virus es igual a la probabilidad de un brote epidémico en una red de intercambio de archivos, suponiendo que un pirata informático esté explotando la vulnerabilidad. Es de suponer que los ciberdelincuentes involucrados necesitaban cargar constantemente archivos maliciosos en un rastreador de torrents, creando más fuentes de infección. El tiempo de creación y carga de un archivo de tamaño promedio es de 0.4 horas. Por lo tanto, hasta que se repare la vulnerabilidad, se pueden cargar 951.14 archivos maliciosos. En condiciones normales de trabajo, un hacker puede crear 0.3 epicentros de infección por hora. A partir de esto se estima que la probabilidad de infección por este medio es del **78%**.
- ***Correo electrónico***: en este caso se debe considerar la probabilidad de que un mensaje malicioso traspase los filtros de spam del servidor de correo, y la tasa de detección promedio de un programa antivirus instalado en la PC del usuario, entre otros parámetros. Como resultado, se obtuvo que la probabilidad final de ser infectado por correo electrónico es del **1%**.
- ***Sitios web no autorizados:*** Las descargas automáticas son una forma común de tales ataques, La probabilidad de este tipo de infección no puede estimarse analíticamente, por lo se realizó un muestreo y en base a los resultados del mismo se estipulo que la probabilidad de estos ataques es del **30%**.
- ***Phishing o siembra:*** este es el tipo de ataque más exitoso. Para analizarlo se tuvo en cuenta la probabilidad de que una carga maliciosa atraviese el antivirus, y la tasa de detección promedio. Como resultado de obtuvo que la probabilidad de un ataque exitoso es aproximadamente del **60%**.
- ***Ataque directo a la red interna:*** Un ataque directo a la red interna de una organización objetivo depende de la topología y la infraestructura de la red. El tiempo para comprometer un modelo de proceso aleatorio es un compuesto de acciones de ataque dirigidas a la explotación de vulnerabilidades. Suponiendo que el atacante es un experto podemos estimar que la probabilidad de un ataque directo en la red interna de la organización es aproximadamente del **6%**.
- **Como se podrá notar, no se tendrán en cuenta las rutas correspondientes a los dispositivos móviles.**

MODELO DE SIMULACIÓN

$$M = (X, Y, S, \delta int, \delta ext, \lambda, ta)$$

Donde

$\Longrightarrow$ X = {IAF, IAT}

$\Longrightarrow$ Y = {AE, AF}

$\Longrightarrow$ S = {Activo, Inactivo, ADR, AM, ADPC, SWT, CE, SWNA, P}

$\Longrightarrow$ δint = {Activo = δint(Inactivo), ADR = δint(Activo), AM = δint(Activo), ADPC = δint(Activo), SWT = δint(Activo), CE = δint(Activo), SWNA = δint(Activo), P = δint(Activo)}

$\Longrightarrow$ δext = ESTE NO SUPE COMO REPRESENTARLO

$\Longrightarrow$ λ = {AE = λ(ADR), AE = λ(AM), AE = λ(ADPC), AE = λ(SWT), AE = λ(CE), AE = λ(SWNA), AE = λ(P), ANC = λ(ADR), ANC = λ(AM), ANC = λ(ADPC), ANC = λ(SWT), ANC = λ(CE), ANC = λ(SWNA), ANC = λ(P)}

$\Longrightarrow$ ta = el único que yo supongo que avanza por tiempo es el estado inactivo, que después de x tiempo accionará el estado activo, y este, según un random determinará qué estado será el siguiente. Los demás estados, que representan los ataques, determinaran el tipo de salida que producen según las probabilidades que mencione antes. ¿Cómo hago para representar eso?