ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

INGENIERÍA EN INFORMÁTICA

Proyecto de seguridad informática

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

INGENIERO EN INFORMÁTICA

Proyecto de seguridad informática

- Departamento: Teoría de la señal, Telemática y Comunicaciones
- Director del proyecto: Gabriel Maciá Fernández
- Autor del proyecto: Moisés Gautier Gómez

Granada, 12 de mayo de 2016

Fdo: Moisés Gautier Gómez

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

Estar preparado para la guerra
es uno de los medios más eficaces
para conservar la paz
George Washington

En el siglo XXI toda pasa por ser digital y sino, ya es que lo era antes de llegar a este punto. Quizás muchas de las tecnologías que hoy conocemos se basen en un sistema informatizado, ya sea en su formato de software o sistema embebido. Y es que todo pasa por ser una herramienta software diseñadas para un propósito en concreto: un mecanismo de apertura de puertas mediante tarjetas rfid, procedimientos industriales o aplicados a alguna infraestructura crítica o de bien común, una aplicación del tiempo en tú terminal móvil, el propio sistema operativo con el que se puede leer el documento, etc.

Hay un sinfín de aplicaciones software que hacen nuestro día a día más llevadero y más fácil. Pero hay un punto que no todos conocen y es la necesidad de saber cómo se ha creado ese producto o cómo funciona realmente por su interior. En ése interior, a veces, podemos encontrar cosas que no estaban predestinadas a tener ese comportamiento y debido a ése comportamiento anómalo o imprevisto se generan situaciones de incertidumbre en las que el ser humano debe estar capacitao a afrontar. Dichas situaciones se suelen conocer con el término anglosajón de "bugz sobre los bugs hay una especial categoría que se denominan fallas de seguridad o critical/several bugs.

Estas situaciones contraactuales provocan que nuestro sistema, sea el que fuese, actúe de forma inesperada ante un input de información permitido o legítimo permitiendo un uso inadecuado de los recursos a los que se acceden mediante la aplicación. De este concepto o problema, surge en gran medida, el término de seguridad informática el cuál intenta abarcar y dar solución a estos problemas que pueden ir desde un simple fallo de desarrollo a un fallo crítico que comprometa la seguridad o confidencialidad de los documentos de una empresa o gobierno.

Debido a esta problemática, surge la necesidad de analizar, monitorizar y generar sistemas de seguridad perimetral que permita a las empresas ver que tipo de tráfico interno

se genera, que tipo de tráfico externo tiene y cómo se hace uso de él (navegación hacia el exterior, tuneles vpn, conexiones remotas a dispositivos, etc). Así pues, se podría decir que para obtener este tipo de eventos sobre protocolos, tráfico, dns, ips, vpns,.. se tienen que configurar dispositivos de seguridad para la recolección de estos tipos de inputs o fuentes.

Una de las fuentes más conocidas dentro del mundo de la informática es el firewall, pero no así de las de un uso más extendido dentro del mundo doméstico sino el comercial o corporativo. Y dentro de los muchos tipos de softwares enfocados a tráfico (firewall) se encuentra el paquete de las distribuciones GNU/Linux iptables (software fruto del proyecto Netfilter para el kernel de GNU/Linux). Con esta herramienta se pueden definir políticas de filtrado de tráfico para cualquier tipo de protocolo tcp/udp que queramos limitar entre el exterior y nuestra máquina y viceversa. Además, estas políticas nos permiten derivar dicho tráfico a archivos que podemos manipular obteniendo así los eventos que representan al tráfico generado por una máquina conectada a una red, que posteriormente podemos manipular para generar estadísticas o tipos de uso para una red.

Aquí nace éste proyecto final de carrera. La solución que se busca desarrollar se denominaría Sensor de Seguridad (Security Sensor). De forma muy general, éste software se encargará de monitorización información de una máquina (logs, red (iptables), etc), almacenarla, visualizarla y además realizar un procesado de la misma con un algoritmo de obtención de características específico.

1.1. Objetivos

El objetivo principal del proyecto es controlar los tipos de conexiones entrantes y salientes de una máquina en sus diferentes protocolos de comunicación y mecanismos de gestión.
La aplicación deberá cumplir los siguientes requisitos:

- Ser una herramienta multiplataforma y que permita a cualquier usuario definir sus propias interfaces de gestión de eventos.
- Dotar de funcionalidad gráfica que permita extraer información en tiempo real con gráficas o mecanismos visuales (en web) del sistema de base de datos que ha procesado los inputs de las fuentes para las que ha sido configurada.
- Dotar de una api interna que nos permita extraer información en tiempo real en un formato uniforme para la web o para que cualquier usuario pueda usar la funcionalidad del proyecto para su propio beneficio usando herramientas generadas en el back-end para otro tipo de aplicaciones.
- Ser parte de un todo, en el que el todo sea un SIEM capaz de obtener información de las diferentes sondas o módulos, que en este caso, sería la solución desarrollada.

- Desarrollar una sonda para procesar eventos logs de firewall.

1.2. Contexto: Historia sobre los sistemas de detección

[Breve explicación de la historia de los sistemas de detección]

1.3. Alcance

El proyecto Security Sensor dará cómo resultado una aplicación software con interfaz visual web, que cumplirá con todos los objetivos y especificaciones indicados en el apartado anterior.
La aplicación se distinguirá en varias partes:

- Configuración y parametrización de los eventos logs de firewall en el dispositivo.
- Recolección de logs mediante rsyslog y su posterior procesamiento.
- Almacenaje en BD para su posterior manipulación interna o externa (api).
- Extracción de características de los eventos y visualización de los mismos mediante un servicio web.
- Aplicación de un algoritmo de procesamiento para comprimir la información en formato módulo para ser servida a un nodo central de gestión (SIEM).

[PONER AQUÍ ESQUEMA DEL WORKFLOW DE PROCESAMIENTO: lo de los assets y demás]

1.4. Visión global

En cuanto a la estructura de esta memoria del proyecto final de carrera, tras éste capítulo dónde se presentan los objetivos y la visión en general del proyecto, se expone el estado del arte y el análisis de requisitos previos al desarrollo software.
En el capítulo siguiente, veremos la etapa del diseño de software así cómo posterior evaluación del mismo.
Finalmente, se presentan las conclusiones generales obtenidas una vez realizado el proyecto, así también la planifiación del mismo y estimación de costes.
Además, se presentan las referencia bibliográficas dónde se incluyen las fuentes consultadas para la elaboración de este proyecto, un resumen que engloba las generalidades fundamentales de la aplicación, una guía de utilización (manual de usuario), una guía de instalación, un compendio del software utilizado para el desarrollo y otro de los lenguajes de programación, y finalmente, la licencia completa del documento.

Capítulo 2

Estado del arte

Si por estado del arte se refiere a los precedentes del proyecto cómo herramienta, pues hay muchas que ya hacen algo similar cómo lookwise, logstash, elasticsearch, kibana, etc.
que aportan estos software al público general y porque la necesidad de mi pfc tal cuál.
Podría explicar que se enmarca en un proyecto de investigación del grupo o invertarme una nube de humo muy grande.
Realizar una aproximación temporal en el tiempo para describir la evolución de las otras soluciones. Esto no lo entiendo muy bien [NOTA]
Tecnologías (descripción) python: librerias cómo pygtail, gunicorn, django (framework), d3, c3, react

Capítulo 3

Especificación y análisis de requisitos

Análisis de requisitos en el plan se necesita desarrollar una aplicación para solventar éste problema y se ha planteado con estas premisas (source, controller, manager, etc) que a su vez han necesitado de un aprendizaje, explicar porque se ha hecho una cosa de una manera (sqlite3 vs mysql), etc.

Especificacion de los requisitos
Análisis (Justificacion de la especificacion)

Capítulo 4

Diseño

<diseño>——</diseño>

- estructura de clases (diagrama)
- diseño de la vista (patrón)
- arquitectura del sistema

Capítulo 5

Implementación

<implementacion>——</implementacion>

- git
- archivos de instalación
- archivos de configuración (estructura)
- fuentes del código
- jerarquía de clases (implementación)

Capítulo 6

Evaluación

- pruebas funcionales y no funcionales
- caja blanca
- caja negra

Capítulo 7

Planificación y estimación de costes

Ya tengo hecho así por encima el diagrama de gantt de todo el proyecto. Tendría que definir las tareas en el diagrama pero lo que haré será meter aquí los apuntes que he ido metiendo en la herramienta Taiga para que se vea alguna descripción o apuntes que he ido tomando por cada tarea del diagrama.

7.1. Software utilizado

El lenguaje usado es python 2.7, luego django con todo los paquetes asociados (poner algún enlace de referencia dónde se listen los paquetes más importantes que he tenido que instalar sin nombrar todas las dependencias de estas)
He usado cómo IDE Pycharm del projecto jetbrains
Para la memoria LaTeXy para las gráficas tikz supongo.

7.2. Licencia

La licencia del proyecto para su posterior uso. En el actual repositorio de bitbucket no se ha especificado la licencia tal cuál, pero cuando ya este más estable el asunto también lo pondre en github para subirlo y demás. Ahí ya si que tendré que especificarla. En principio sino es para ningún propositio comercial, con la MIT sería suficiente. Sino alguna variante de la GPL o la Mozilla o similares.