Universidad Carlos III de Madrid

Escuela Politécnica Superior

PROYECTO FIN DE CARRERA

SISTEMA DE INTERCEPTACIÓN Y ANÁLISIS DE COMUNICACIONES

Autor:
Carlos Gacimartín García

Tutor:
José Alberto Hernández Gutiérrez

Leganés, Enero 2009

Índice

AGRADECIMIENTOS

Expresar los agradecimientos por su ayuda en este proyecto a:

- José Alberto Hernández Entierre, Universidad. Carlos III, Madrid.
- Gianluca Costa y Andrea De Franceschi, proyecto Xplico.org.
- A la comunidad GNU en general, por el desarrollo del software libre que han hecho posible este proyecto, y su gran disponibilidad a ayudar y colaborar.

RESUMEN

Se pretende diseñar una solución hardware y software para facilitar el análisis de tráfico de una conexión a Internet por las fuerzas de seguridad sin intervención ni conocimiento necesario del ISP.

El tráfico generado por un usuario en Internet se intervendrá y analizará, y según unos baremos se ponderará para crear alarmas que indiquen una actividad sospechosa, de cara a requerir mayor atención por parte de las autoridades.

Para ello se planteará el hardware y software necesarios, desarrollándose además el núcleo o core del software con facilidades para que pueda mantenerse y ampliarse.

Mediante la captura de todo el tráfico de red de un individuo, se realizarán en tiempo real y solapándose las siguientes tareas:

1.- Realizar un primer análisis de la información generando alertas ante determinados contenidos detectados. (**nivel INDECT WP3**)

2.- Almacenar todo el tráfico de red capturado y reenviarlo a un servidor propio.

3.- Realizar un análisis más detallado mediante la decodificación completa de los protocolos deseados (nivel INDECT WP4) utilizando para facilitar su estudio por un operador.

Debido a la magnitud de este proyecto, se planifica su desarrollo para tenerlo listo y en funcionamiento en el evento deportivo “Eurocopa de Polonia **2012**”.

En este **documento** se plantea el **diseño completo** de la solución y se profundiza exclusivamente en el **módulo inicial de detección de ficheros** sospechosos.

Se estiman posteriores ediciones de este documento para completar la información y diseños existentes.

Capítulo 1

Introducción

1.1 INDECT *WP*3

El proyecto INDECT, financiado por la Unión Europea, tiene entre sus objetivos principales desarrollar una plataforma de monitorización de la información en Internet, centrándose en la búsqueda de amenazas. Para ello realiza captura de datos multimedia y procesamiento automático de información para reconocer comportamiento anormal o violencia.

Los objetivos principales del proyecto INDECT son:

- Desarrollar una plataforma para el registro y el intercambio de datos operativos, la adquisición de contenidos multimedia, procesamiento inteligente de toda la información, la detección automatizada de amenazas y el reconocimiento de comportamiento anormal o la violencia.

- Desarrollar el prototipo de una red integrada, centrada en el apoyo a las actividades operacionales de los agentes de policía, proporcionando técnicas y herramientas para la observación de varios objetos móviles.

- Desarrollar un nuevo tipo de motor de búsqueda que combine la búsqueda directa de imágenes y de vídeo basado en el contenido de marcas de agua, y el almacenamiento de los metadatos en forma de marcas de agua digitales.

- Desarrollar un conjunto de técnicas de apoyo a la vigilancia de los recursos de Internet, análisis de la información adquirida, y la detección de amenazas y actividades criminales.

Los principales resultados esperados del proyecto INDECT son:

- Realizar una instalación de prueba del sistema de supervisión y vigilancia en diversos puntos de la aglomeración de la ciudad y demostración del prototipo del sistema con 15 estaciones nodo.

- Implementación de un sistema de computación distribuida que sea capaz de adquirir, almacenar e intercambiar bajo demanda datos, así como su procesamiento inteligente.

- Construcción de prototipos utilizados para el seguimiento de objetos móviles.

- Construcción de un motor de búsqueda para la detección rápida de personas y de documentos basados en la tecnología de marcas de agua y aprovechamiento de la amplia investigación de la tecnología de marcas de agua utilizada para la búsqueda semántica.

- Construcción de agentes asignados a la vigilancia continua y automática de los recursos públicos, tales como: páginas web, foros de discusión, grupos de Usenet, servidores de archivos, las redes P2P, así como los sistemas informáticos individuales

- Elaboración de un sistema de recolección de inteligencia basado en Internet, tanto activa como pasiva, y demostrando su eficiencia en una forma cuantificable.

Indect tiene un comité ético, formado principalmente por la Policía Norirlandesa, para hacer saber a los participantes cuales son los objetivos en este ámbito, estableciendo qué barreras no se pueden sobrepasar.

Técnicamente no se crea ninguna fuente de información nueva, si no que se tiene como objetivo principal el aprovechar al máximo las fuentes existentes (Internet, cámaras de video-vigilancia, etc).

Además, uno de los objetivos primordiales es conseguir la máxima mecanización del procesamiento de información, de modo que se elimine el factor humano de la subjetividad. Además, esto implicará menos personal necesario (menor posibilidad de mal-utilizar la información existente con otros fines distintos de los originales), y los agentes y fuerzas de seguridad nacionales podrán centrarse en su labor real.

También permitirá extender las investigaciones de sospechosos o acusados a los individuos que, mediante la información contrastada que se saque de este sistema, se demuestre que están implicados en el mismo crimen. Por ejemplo, si este sistema detecta el tráfico de imágenes de pederastia, no sólo se informará del emisor si no de todos los posibles receptores para poder extender la investigación.

Es importante señalar también que un sospechoso indicado por el sistema Indect será reportado a las autoridades pertinentes, no se le acusará de nada directamente. Además, no se almacenarán datos relevantes al mismo a no ser que haya razones legales para ello.

Para la ejecución de Indect en cada país será necesario estudiar la legislación propia. Además, será plenamente compatible con la Carta Europea de los Derechos Fundamentales y el Acta de Protección de Datos (1998).

1.2 OBJETIVOS DEL PRESENTE PROYECTO

Este documento pretende dar cobertura al nivel INDECT WP3, tareas 1ª,2ª y 4ª asignadas a la Universidad Carlos III:

- WP3 - Tarea 1ª: Elaboración de la estructura lógica de INDECT-MAS: Proceso de monitorización, subsistemas de toma de decisión y especificaciones preliminares de los roles de usuario y posibles interacciones.
- WP3 - Tarea 2ª: Diseño de una estructura basada en componentes de los agentes y la plataforma prevista, así como de la interface. Uso de metodologías de diseño e ingeniería de software.
- WP3 - Tarea 4ª: Prototipo de agente de monitorización , representación de la misma y uso de bases de datos o lenguajes de descripción de ontologías. Valoración del incremento de la información, lo cual puede implicar la reorganización de las estructuras de la información.

1.3 MÉTODOS ACTUALES / STATE OF ART

Actualmente todos los países medianamente evolucionados disponen de tecnologías, en mayor o menor medida, para la interceptación de información. No obstante, y debido a la discreción requerida para que sean difícilmente combatibles o eludibles por el crimen, apenas hay vagas referencias o confirmaciones oficiales de estos sistemas.

Principalmente, y sin poder ahondar en ellas por falta de información pública, caben destacar:

- **Echelon**: es considerada la mayor red de espionaje y análisis para interceptar comunicaciones electrónicas de la historia. Controlada por la comunidad UKUSA (Estados Unidos, Canadá, Gran Bretaña, Australia, y Nueva Zelanda), ECHELON puede capturar comunicaciones por radio y satélite, llamadas de teléfono, faxes y e-mails en casi todo el mundo e incluye análisis automático y clasificación de las interceptaciones. Se estima que ECHELON intercepta más de tres mil millones de comunicaciones cada día. A pesar de haber sido con el fin de controlar las comunicaciones militares de la Unión Soviética y sus aliados, se sospecha que en la actualidad ECHELON es utilizado también para encontrar pistas sobre tramas terroristas, planes del narcotráfico e inteligencia política y diplomática. Sus críticos afirman que el sistema es utilizado también para el espionaje económico y la invasión de privacidad en gran escala. La existencia de ECHELON fue hecha pública en 1976 por Winslow Peck.

- **Carnivore**: Este software, utilizado por el FBI, se instala en los proveedores de acceso a Internet y, tras una petición proveniente de una instancia judicial, rastrea todo lo que un usuario hace durante su conexión a Internet. En teoría tiene capacidad para discernir comunicaciones legales de ilegales. Dispone de capacidad “quirúrgica” de distinguir entre sujetos interceptados y no interceptados y posibilidad de distinguir entre datos interceptables de un sujeto y datos no interceptables, basándose en los poderes concedidos por la orden judicial de intercepción.

- **Sitel**: Sistema de escuchas telefónicas del Ministerio de Interior de España utilizado por la Policía Nacional, la Guardia Civil y el Centro Nacional de Inteligencia (CNI). También es conocido como Sistema Integrado de Interceptación de Telecomunicaciones, Sistema Integrado de Interceptación Legal de Telecomunicaciones y Sistema Integral de Interceptación de las Comunicaciones Electrónicas.

- **SitCen**: Punto de encuentro policial de la Unión Europa, orientado a monitorizar casos de influencia común como las armas de destrucción masiva. Desde Febrero de 2005 mantiene también una unidad antiterrorista, y mantienen una supervisión continua de la información.

- **ADABTS**: "Detección automática de comportamientos anómalos y amenazas en espacios concurridos "), un sistema de bajo coste para la vigilancia activa con la finalidad de detectar posibles comportamientos anormales en espacios concurridos.

1.4 PROBLEMÁTICA ACTUAL

Cada día el criminal va haciendo más uso de la red frente a otros medios de comunicación. Así mismo se producen delitos que sólo ocurren en Internet. Para dar soporte a ambas problemáticas, se han creado sistemas como Echelon para poder acceder a esa información que al ser transmitida deja de estar bajo la ocultación y protección del delincuente, creándose una oportunidad para las fuerzas y cuerpos de seguridad de poder utilizarla para combatir la criminalidad.

Existen sistemas de los cuales se conoce poca o nada a parte de su existencia. En este proyecto, auspiciado por la Unión Europea y por tanto por aquellos que conocen las tecnologías existentes, se propone una nueva generación de tecnologías para obtener más información de los medios actuales y posibilitar el cruce de la misma, lo cual aumentaría la eficacia de las policías europeas y consecuentemente la seguridad del ciudadano.

1.5 DIFICULTADES

Ante la complejidad del proyecto, se presentan los siguientes escollos a salvar:

1. **Que el sospechoso no detecte el sistema de espionaje**: dentro del espectro de criminales y organizaciones susceptibles de ser monitorizadas, hay que emplear un sistema que no levante sospecha tanto a usuarios tecnológicamente normales como a sospechosos con habilidades de contraespionaje electrónico.
2. **Que el ISP no se lo detecte**: al poder realizarse la instalación en secreto, al margen del ISP, sin su consentimiento o autorización, pero siempre con autorización judicial, se debe realizar de manera que no puedan detectar el sistema ni les cause desbarajustes en su modo normal de operación, que les llevase a investigar el estado de la línea con la que se transmiten los datos.
3. **Que se pueda procesar todo el tráfico sin perderse nada**: todo el tráfico generado debe ser monitorizado, sin excepción.
4. **Que funcione las 24 horas, 365 días al año**: si bien no se prevén intervenciones tan largas, se debe disponer de una infraestructura capaz de soportar estas cuotas de cara a garantizar altos niveles de capacidad y criticidad puntuales.
5. **Que no se escapen datos sospechosos**: es preferible una falsa alerta a perder información valiosa o crítica.
6. **Que las alertas lleguen a tiempo**: el procesamiento de la información debe realizar a la mayor celeridad posible, deseable en tiempo real.
7. **Que exista validez legal de las pruebas**.

Capítulo 2

Etapas y planificación

2.1 Planificación inicial. Ciclo de vida seleccionado

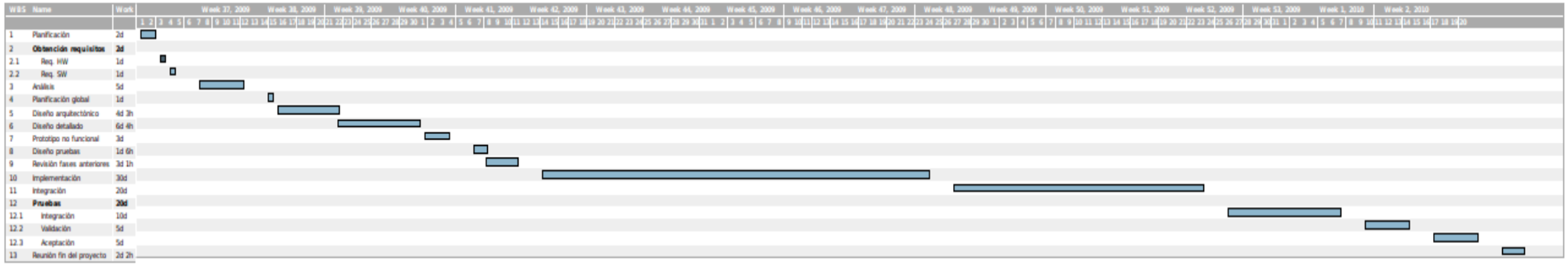
Se plantean las siguientes Etapas:

- Planificación: obtención de requisitos generales, acuerdo con el jefe de equipo y modelado final de los requisitos.
- Obtención de requisitos HW: etapa de evaluación de las necesidades existentes hardware.
- Obtención de requisitos SW: etapa de evaluación de las necesidades existentes software para conocer los casos de uso y requisitos completos de la aplicación.
- Análisis: se realiza un diseño global de la solución.
- Planificación global: conocidos los requisitos de SW y HW, y haciendo las estimaciones propias de tiempo de cada desarrollo, se coordinan las mismas para programar una integración en el tiempo.
- Diseño arquitectónico: Diseño a alto nivel de los componentes software y hardware que intervendrán en el proyecto.
- Diseño detallado: Diseño de cada componente software en base a los requisitos obtenidos.
- Diseño de pruebas general: conociendo a alto nivel la arquitectura y funcionalidades, plantear una batería de pruebas amplia y generalista.
- Prototipos: generación de prototipos para depurar los requisitos.
- Diseño de pruebas de validación, integración y aceptación: diseño de las pruebas a ejecutar en cada fase.
- Revisión fases anteriores: iteración en las fases anteriores para que, partiendo de una visión completa del proyecto, revisar y mejorar todo lo posible, fundamentalmente planificaciones y diseño.
- Implementación: codificación de los elementos proyectados.
- Integración.
- Pruebas de validación: Ejecución y validación de las pruebas de validación del sistema..
- Pruebas de aceptación: Ejecución y validación de las pruebas de aceptación del prototipo final generado por parte del cliente (INDECT)
- Reunión fin del proyecto: evaluación del proyecto, obtención de feedback de la ejecución en laboratorio y establecimiento de líneas futuras.

xplicoAlerts

Company: uc3m
Manager: Carlos Gacimartín
Start: September 1, 2009
Finish: January 20, 2010
Report Date: August 31, 2009

Gantt Chart



Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost
1	Planificación	Sep 1	Sep 2	2d			0%
2	**Obtención requisitos**	Sep 3	Sep 4	2d			
2.1	Req. HW	Sep 3	Sep 3	1d			100%
2.2	Req. SW	Sep 4	Sep 4	1d			0%
3	Análisis	Sep 7	Sep 11	5d			0%
4	Planificación global	Sep 14	Sep 14	1d			0%
5	Diseño arquitectónico	Sep 15	Sep 21	4d 3h			0%
6	Diseño detallado	Sep 21	Sep 29	6d 4h			0%
7	Prototipo no funcional	Sep 30	Oct 2	3d			0%
8	Diseño pruebas	Oct 5	Oct 6	1d 6h			0%
9	Revisión fases anteriores	Oct 6	Oct 9	3d 1h			0%
10	Implementación	Oct 12	Nov 20	30d			0%
11	Integración	Nov 23	Dec 18	20d			0%
12	**Pruebas**	Dec 21	Jan 15	20d			
12.1	Integración	Dec 21	Jan 1	10d			0%
12.2	Validación	Jan 4	Jan 8	5d			0%
12.3	Aceptación	Jan 11	Jan 15	5d			0%
13	Reunión fin del proyecto	Jan 18	Jan 20	2d 2h			0%

Resources

Fig 1 – Planificación temporal de la primera fase.

En el nivel de **Implementación**, se pueden **desgranar** las siguientes tareas:

TAREAS	HORAS
ETAPA 1	
Diseño teórico del modelo completo.	100 h
Prototipo de captura de información.	100 h
Prototipo de decodificación de la información intervenida.	200 h
Prototipo de detección de ficheros sospechosos.	200 h
Prototipo de generación de alarmas.	200 h
Pruebas, validación y recolección de feedback.	184 h
TOTAL	**984 h**
ETAPA 2	
Hashes relacionales por DeepToad.	200 h
Prototipo de detección de palabras sospechosas.	600 h
Prototipo de detección de URL's sospechosas.	200 h
TOTAL	**800 h**
ETAPA 3	
Configuración	50 h
Estadísticas de rendimiento.	100 h
Testeo intensivo de los prototipos.	150 h
Apoyo al proyecto Xplico en el soporte a nuevos protocolos a decodificar.	400 h
TOTAL	**700 h**
ETAPA 4	
Mejoras de rendimiento: threads.	50 h
Mejoras de rendimiento: ext4.	25 h
Mejoras de rendimiento: PF_RING	100 h
Mejoras de rendimiento: Monit.	100 h
TOTAL	**275 h**
ETAPA 5	
Envío de alertas por emails.	50 h
Duplicación de la información por software: IPTABLES	50 h
API para interconexión con otros sistemas de Indect.	100 h
TOTAL	**200 h**
ESTIMACIÓN TOTAL	**2959 h**

Siendo la **primera etapa la correspondiente a este proyecto**, y proyectándose las siguientes para el desarrollo en los siguientes años.

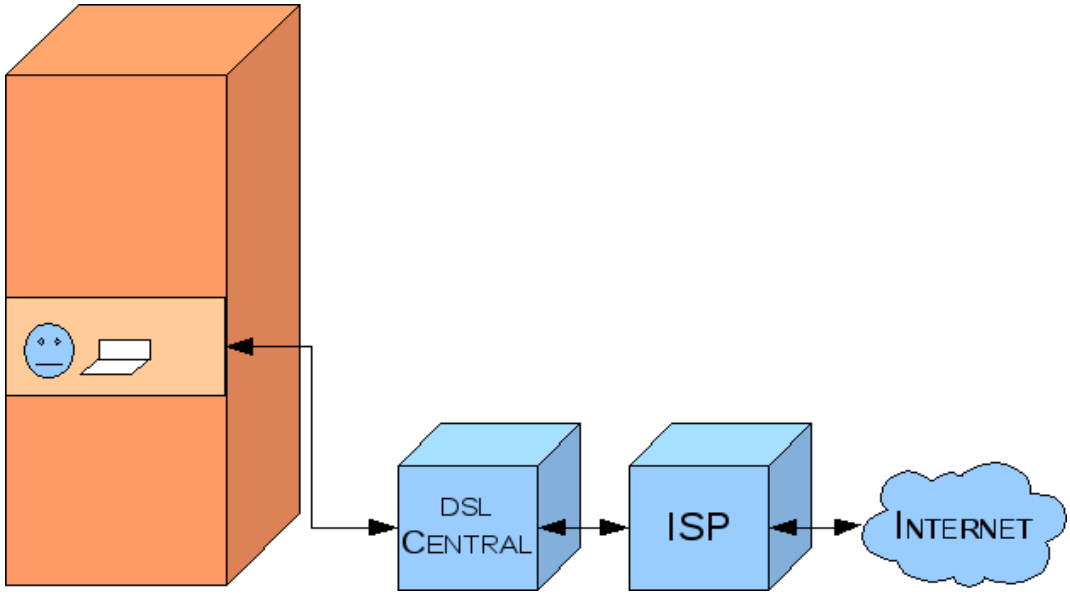
Además quedan especificados todos los requisitos en el “ Anexo VII, requisitos completos”.

Capítulo 3

Análisis

3.1 Arquitectura preliminar y análisis de las tecnologías impuestas

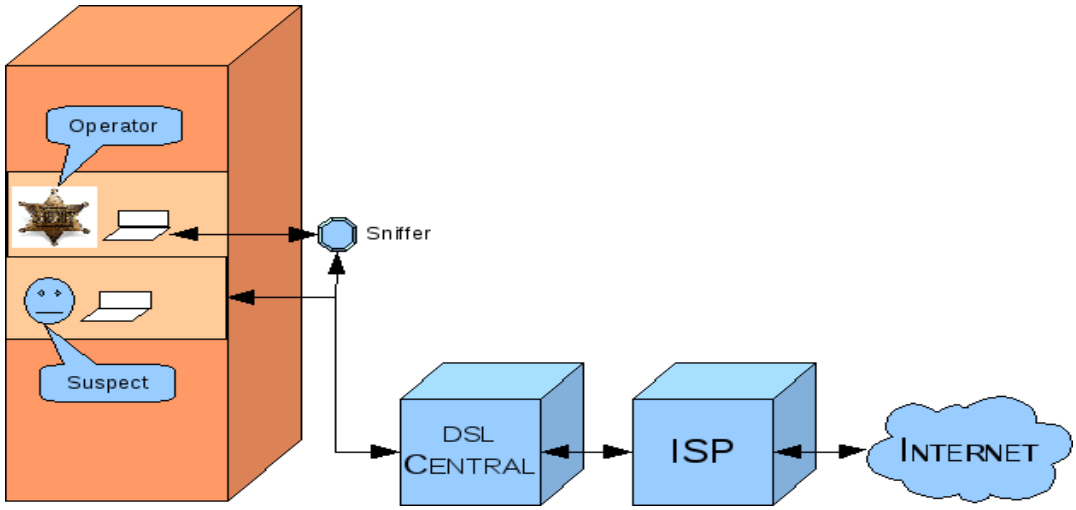
La finalidad principal de este proyecto es poder interceptar los datos de una línea de Internet, por ejemplo ADSL, y poder analizarlos. Para ello, hay que tomar las medidas necesarias de cara a no ser descubierto el propio sistema. Por ello, se plantea un escenario normal como el siguiente:



En el que un ciudadano sospechoso realiza una actividad en Internet. Sus datos viajan desde sus equipos hasta la central DSL de su demarcación, que reenvía la información al ISP y de ahí ya salen a Internet.

Si se requiere que la monitorización del usuario no sea conocida por el ISP, **evitando así cualquier filtración**, no se puede en principio intervenir las áreas de "Central DSL" y mucho menos "ISP". Es necesario monitorizar al usuario desde la zona menos vigilada, el bucle de abonado o cableado tendido desde el domicilio del usuario hasta la central DSL. No obstante el sistema es compatible con la instalación en la propia instalación DSL (DSLam) ó en el ISP, tanto para monitorizar a un usuario como a grupos de ellos.

Por ello se propone una intervención de la línea como la siguiente:

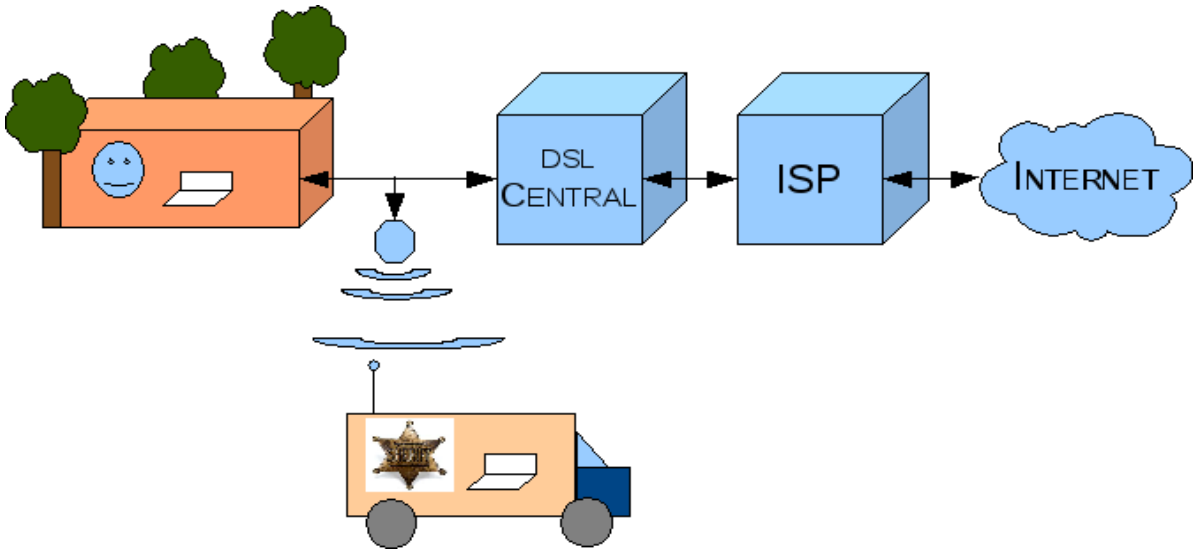


En este gráfico se presenta la misma instalación DSL del usuario existente alterada con un nuevo elemento que interceptará sus comunicaciones con el fin de duplicarlas y re enviárselas a un operador autorizado judicialmente para realizar esta monitorización.

La instalación consistiría en intervenir el par de cobre o línea telefónica que proporciona DSL al usuario espiado con un elemento que copie las señales que viajan por ese cable, sin filtrarlas en ningún caso. Además sería necesario establecer algún tipo de conectividad entre el sistema y el operador.

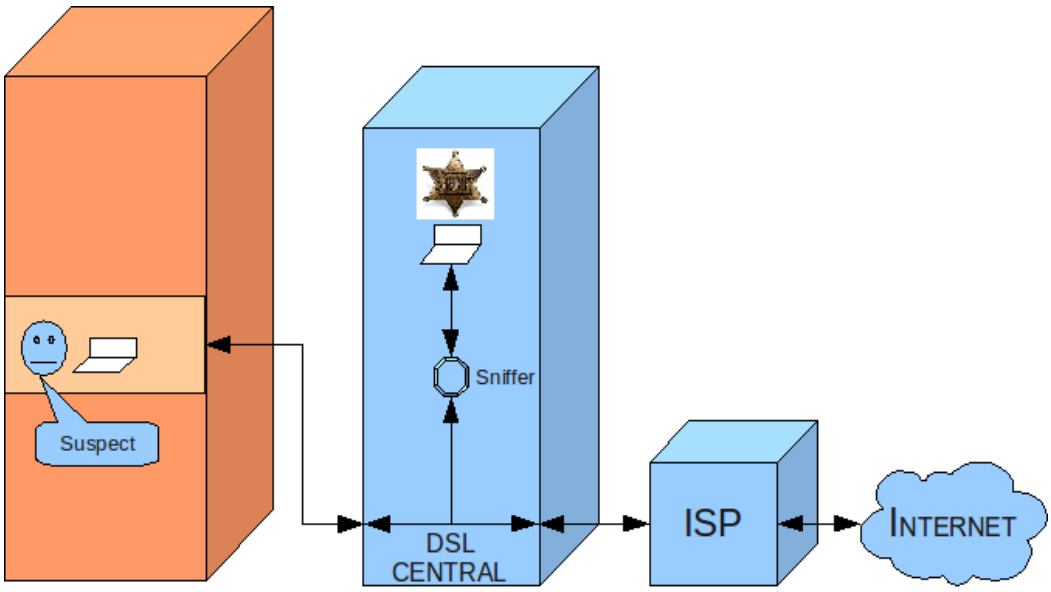
Este sistema interceptador de las señales DSL debe estar lo más próximo a la línea telefónica para no generar una pérdida de señal (SNR).

Otro ejemplo, interviniendo una finca rural, podría implicar el uso además de tecnologías inalámbricas:

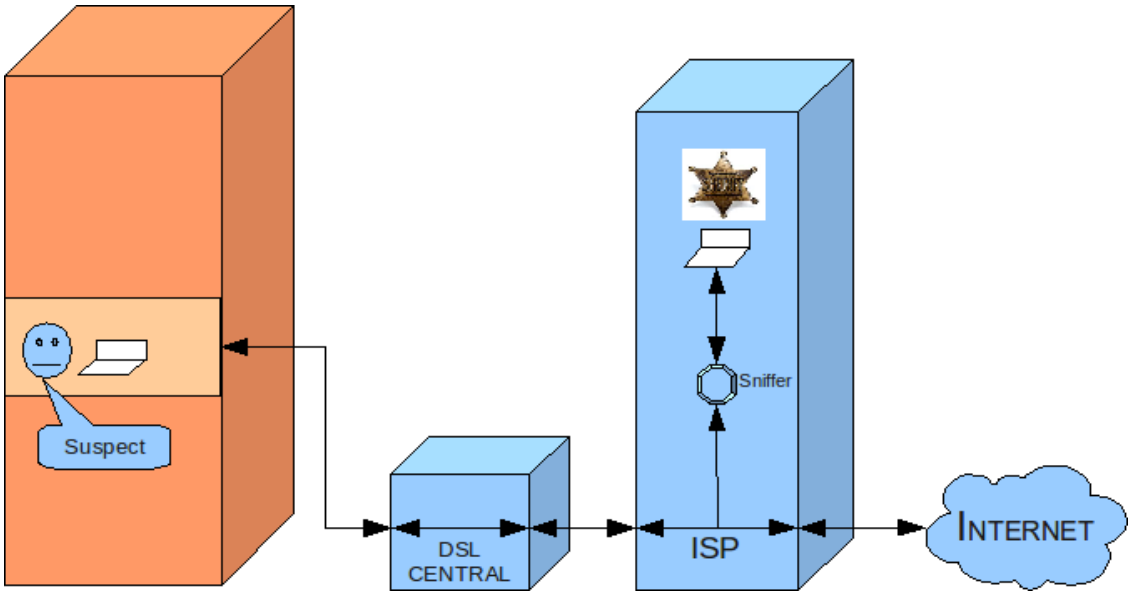


Al no disponerse de conectividad cableada directa con el sistema de intercepción, por discreción, se recurre al uso de tecnologías inalámbricas. Éstas pueden ser uso de wifi ó 3G.

La tercera disposición implicaría la instalación del sistema en el DSLam o centralita DSL.

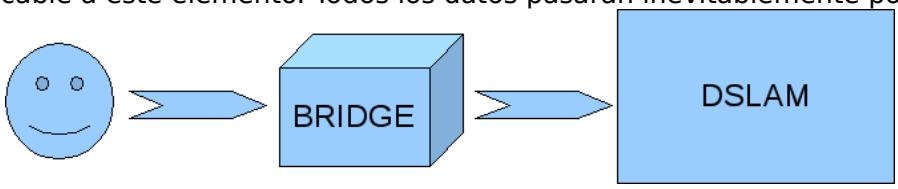


La tercera disposición implicaría la instalación del sistema en el proveedor de acceso a Internet:



Entrando en detalle en el sistema, es necesario determinar su tipología y hacer la elección más segura basando el criterio en no interferir ni crear problemas al usuario monitorizado, de modo que éste no perciba una alteración, bajada de rendimiento o pérdida de conectividad que le lleve a pensar que está siendo monitorizado y/ó que implique abrir una incidencia técnica en el ISP, el cual detectaría el dispositivo. Se plantean las siguientes arquitecturas:

1. **Bridge**: la línea telefónica se corta físicamente, y se conecta cada nuevo extremo del cable a este elemento. Todos los datos pasarán inevitablemente por él.



Actualmente existen dos tipos de bridges:

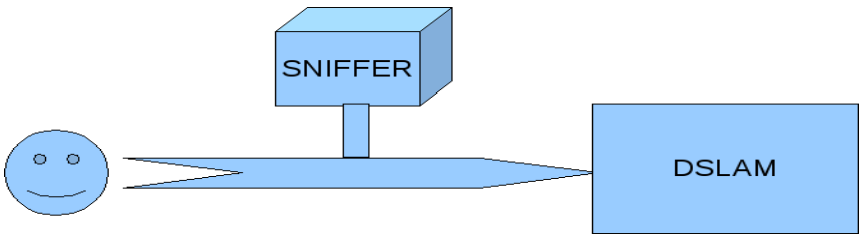
- **Bridge hardware**: dispositivo electrónico que copia los datos de une extremo en el otro y viceversa, siempre y cuando entienda que esos datos viajan en ese sentido de tráfico.

- **Bridge software**: dispositivo compuesto de hardware y software con las mismas funciones que el bridge hardware y añadiéndole utilidades como filtrado ó grabación de datos.

Riesgos:

- Bridge software: El uso de este tipo de bridge en un proyecto como éste implica un alto riesgo, al poder producirse un problema software que no permita reenviar ciertas tramas o que genere otras nuevas, y esto cree un problema al usuario monitorizado.
- Incrementaría la latencia, el tiempo empleado en viajar un paquete por la red se vería incrementado.
- Tanto en el bridge software como hardware existe el problema del suministro eléctrico o cualquier problema hardware. Si el bridge no está funcionando, el usuario no tendrá ninguna conectividad con Internet.

2. Sniffer: consiste en una utilidad que captura todas las tramas de datos que llegan hasta uno o varios de sus interfaces. Aplicado a este proyecto, sería útil sacando una derivación de la línea hacia este dispositivo.



Con esta tecnología se presentarían los siguientes riesgos:

- Que la máquina que soporta el sniffer entendiese las tramas capturadas como dirigidas a ella y las respondiese, generando tráfico nuevo. Esto se puede evitar fácilmente mediante hardware, software o ambas.

- Modificación de los parámetros físicos de la línea: atenuación y SNR. La alteración física de la línea puede derivar en un cambio a peor de la velocidad, así como variación de estos parámetros que, de ser conocidos por el usuario monitorizado, le pueden inducir a sospechar.

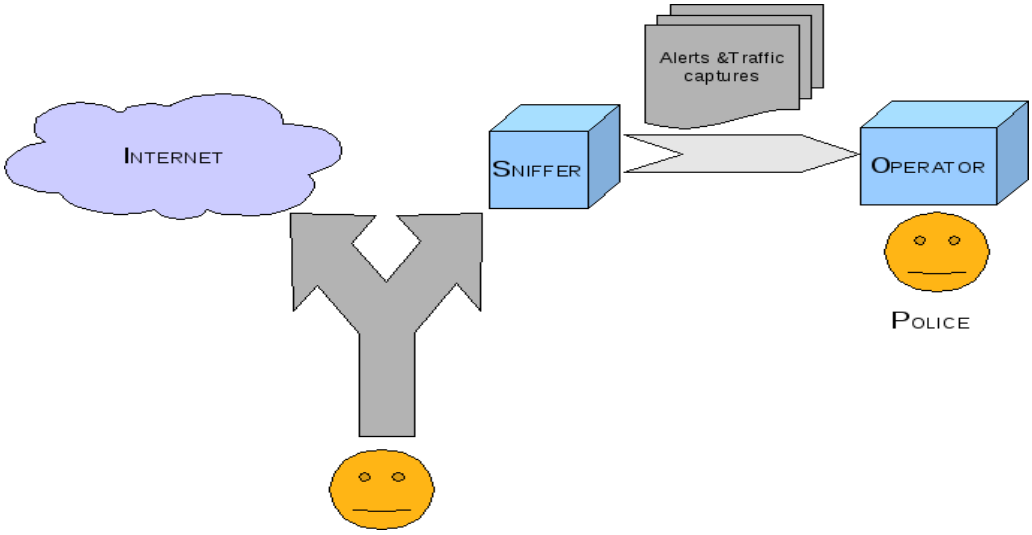
Por todo ello se toma como opción más prudente la arquitectura de sniffer, al ser lo menos intrusiva, así como un hardware de refuerzo que impida la emisión de tramas al canal así como aminore, en la medida de lo posible, el impacto que causa en los parámetros eléctricos de la línea el pincharla con un sistema de espionaje.

Por otra parte, toda la información recolectada por el sniffer tiene que ser procesada automáticamente y en caso de detectar amenazas susceptibles de alerta, avisar al operador. Así mismo, en cualquier caso, todos el tráfico capturado debe ir al Operador para posibilitar su estudio manual. Por ello, se decide procesar el tráfico en el propio sniffer mientras que se va enviando al Operador. De este modo, solapándose ambas partes, se gana tiempo y se generarán, prácticamente en tiempo real, las alertas pertinentes.

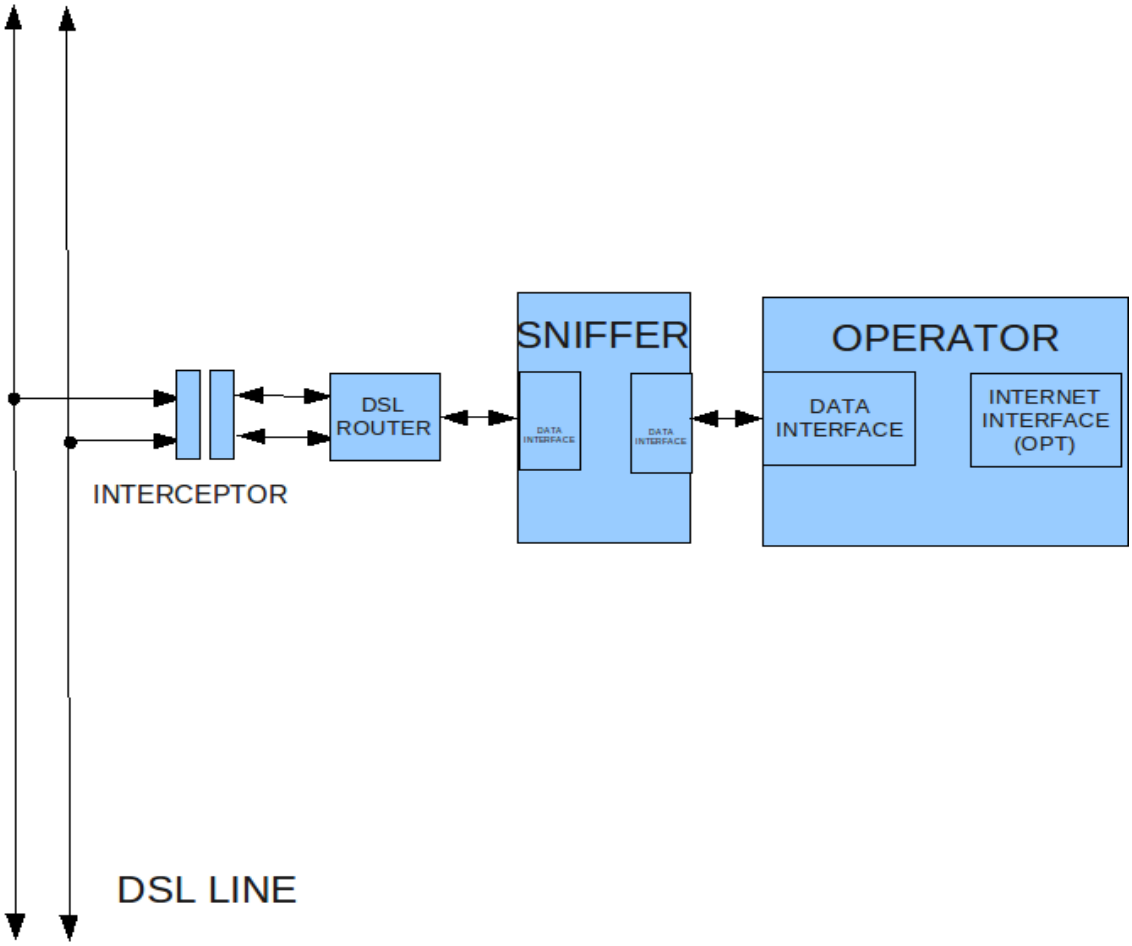
El Operador dispondrá de las capturas de tráfico en su equipo y podrá consultar las alertas tanto en un interfaz web residente en el propio sniffer como en el log local.

3.2 ESTUDIO TECNOLÓGICO

El sistema estará compuesto por los siguientes elementos:



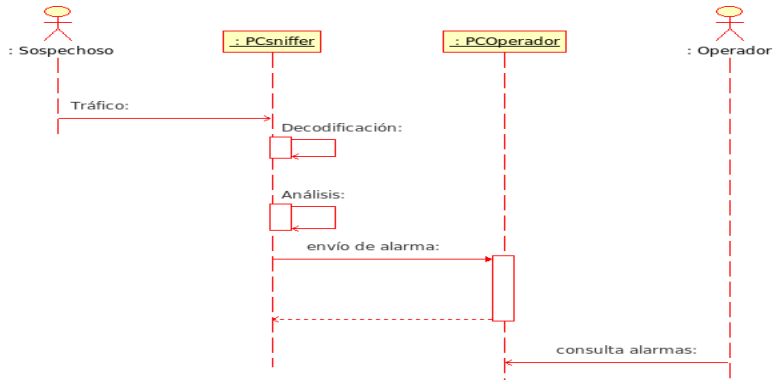
A nivel hardware, se puede analizar el sistema propuesto como el siguiente:



Respecto al procesamiento de la información, se puede realizar en el propio PCSniffer ó en el equipo del Operador. A continuación se evalúan ambas formas:

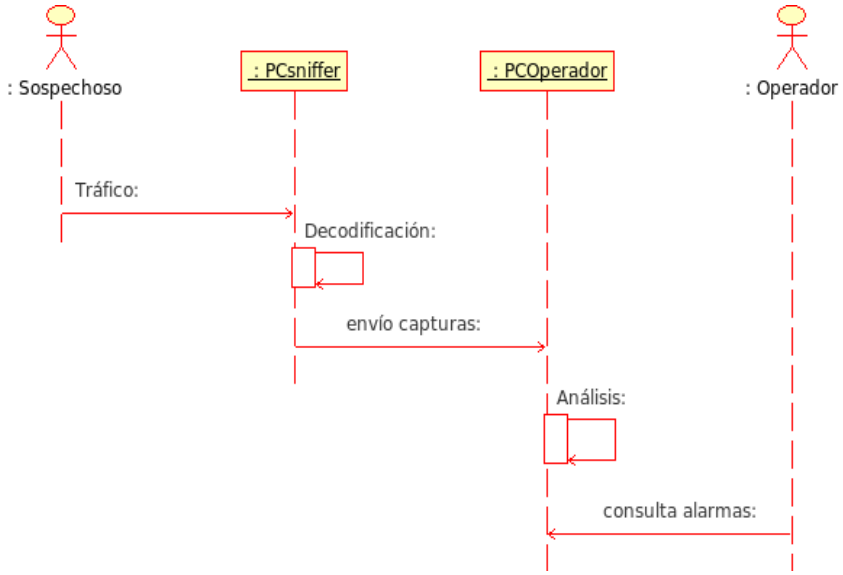
a) Procesamiento de la información en PCSniffer

El propio PCSniffer, que captura las tramas de datos, las procesaría en busca de palabras sospechosas y en caso de encontrar una, generaría la correspondiente alerta. Las capturas reenviarían íntegramente al PCOperador siempre.



b) Procesamiento de la información en PCOperador

El PCSniffer capturaría las tramas, las comprimiría y las enviaría al PCOperador. Allí el software lo procesaría en busca de palabras sospechosas.



Por lo que se plantea el siguiente análisis:

FACTOR	PC-SNIFFER	PC-OPERADOR
Tiempo de generación de alertas	Las alertas se generan en tiempo real.	Hasta que no lleguen las capturas reenviadas, no se pueden analizar y generar alarmas.
Recursos: CPU	Capacidad de cómputo CPU limitada (por espacio y tecnologías disponibles)	CPU ilimitada, al residir en dependencias policiales.
Recursos: HD	Capacidad de almacenamiento (disco duro) reducida	Capacidad de almacenamiento (disco duro) prácticamente ilimitada.

Es preferible disponer del aviso o alerta lo antes posible, por lo que independientemente de los demás factores, se generará el **análisis en el PCSnifer**. En la siguiente página se desarrollarán los requisitos de cada componente.

Así pues, la división resultante distingue 4 componentes:

- **[c001] Interceptor_DSL:** elemento encargado de copiar la información que transcurre por la línea DSL sin alterarla ni levantar sospecha. Se compone de una electrónica sencilla que copie los bits del cable adsl en otro, pero no en sentido inverso.

  Una primera aproximación del diseño de este componente sería:

  [c001.001] Acoplado a la línea: lo que lee la línea causando la mínima pérdida de señal o interferencia posible con la tecnología actual, de cara a ser lo menos invasiva posible.

  [c001.002] Generador de señal: lo que genera las señales en dirección al router. El objetivo es el mismo que si se cortocircuitase la línea, pero **evitando** en todo momento:

  - Modificación de los parámetros eléctricos de la línea (principalmente atenuación)
  - Imposibilitar de cualquier modo que fluya información desde el sistema de espionaje a la línea.

  Nota: no es necesaria la instalación de un microfiltro al procesarse exclusivamente la señal DSL.

- **[c002] Router**: traductor del tráfico de la línea DSL a nivel PPP hacia Ethernet. Conectándole la salida del Interceptor_DSL, convierte los paquetes PPP en tramas IP.

  C002.001 Hardware

  - **C002.001.001 Router**: Router DLS con puerto Ethernet compatible con DSL2+.

  C002.002 Software

  - **C002.002.001 Firmware**: Firmware actualizable y modificable. Es necesario modificar su firmware o driver para que no pueda enviar ningún comando, sólo leer paquetes de la pila TCP/IP.

- **[c003] PCS ó PCSniffer**: decodificador y analizador del tráfico obtenido.

  C003.001 HARDWARE

  - **c003.001.001 Microprocesador**: para el fácil mantenimiento de la aplicación, presentándose una tendencia actual de migración de procesadores de 32 bits a los **64bits**, es necesario elegir estos de cara a no conseguir un temprano desfase del sistema.

  - **C003.001.002 Placa base**: de cara a un fácil mantenimiento del producto, abaratar costes y utilizar tecnología probada y contrastada, es necesario que sea una tecnología extendida y común. Además conviene que sea de un tamaño reducido para su fácil ensamblaje.

  - **C003.001.003, Almacenamiento**: para el almacenamiento de una **gran cantidad de datos**, así como soportar condiciones adversas tales como humedad ó fuertes golpes/**caídas**.

Cálculo de capacidad mínima,

1º) Sistema operativo: Debido a la relativamente pequeña cantidad de espacio requerida en esta parte, se dispondrá en exceso para preveer cualquuier problema o facilitar el crecimiento tecnológico del sistema. Así mismo, en memoria virtual, al ser un sistema crítico que debe funcionar en tiempo real, se dispondrá de hasta 3 veces la memoria RAM instalable (en este caso 8GB de RAM).

CAPACIDAD SISTEMA OPERATIVO	
Sistema operativo base	1 GB
Memoria virtual	24GB
Imprevistos + escalabilidad	4GB
TOTAL	**29 GB**

2º) Aplicación y ficheros temporales: La aplicación manejará tanto capturas de tráfico como información decodificada. La información decodificada se almacenará en el propio PCSniffer para poder ser visualizada comodamente a través del interfaz web. Se hará una estimación defensiva de 50GB

3º) Almacenamiento en caso de fallo: planteando que el según el requisito *"R-REN004"* implica que ante la caída del servidor de almacenamiento es necesario guardar el tráfico de las últimas 24h, se requieren para esta necesidad:

Estimación del máximo tráfico en MB generado por un usuario en un día

Premisas: línea DSL de 100MB, haciendo el máximo uso de ella las 24h, sin comprensión de los datos en HD previo almacenamiento por supuesta saturación de la CPU.

Tráfico por hora = 100MB/seg * 60seg/min*60min/hora = 216.000Mb/hora
Tráfico por día = Tráfico por hora * 24h = 216.000Mb * 24= 5.184.000 Mb/d
Conversión Mb a MB = 5.184.000 / 8 = 648000 MB.
Conversión MB a GB = 648.000MB / 1000 (MB/GB) = **648 Gb**.

Por lo que la capacidad mínima del disco duro del PC_Sniffer debe ser:

CONCEPTO	CAPACIDAD
Sistema operativo	29 GB
Ficheros temporales	50 GB
Almacenamiento en caso de fallo	648 GB
TOTAL	**727 GB**

Luego el sistema podría llegar a una teórica saturación si dispusiese de 549,5 o menos. Por tanto, se incluirá un margen mayor para evitar riesgos no previstos en este análisis, así como para facilitar la operabilidad del sistema en caso de llegar a ese punto.

CONCEPTO		CANTIDAD
Mínimo		727 GB
Riesgo	10,00% del mínimo	72,7 GB
	TOTAL	**799.7 GB**

Además en este componente también hay que premiar la velocidad de acceso, siendo cuanto más alta mejor, ya que se realizarán numerosas lecturas y escrituras continuamente al disco.

- **C003.001.004, Puerto de conexión auxiliar**: preveyendo futuras ampliaciones o necesidades, se incorporará al menos un puerto auxiliar de conexión estándar, cumpliendo los siguientes requisitos:
  - No quedar desfasado en 5 años.
  - Disponer de un ancho de banda de 10MB/seg al menos.
  - Fácilmente mantenible, documentado y apoyado por la industria.

- **C003.001.005, Interfaz de red de entrada**: Interfaz de red para obtener los datos a procesar del usuario monitoreado. Debe ser estándar y disponer de un ancho de banda inicial de 1000MB/s (Gigabit), para facilitar su posterior mantenimiento a la vez que se aumenta el rendimiento del sistema.

- **C003.001.006, Interfaz de red para conexión con operador**. Para posibilitar el envío de las capturas de tráfico, alertas y control de la aplicación por un canal seguro, se dispondrá de una ó varias interfaces para este propósito, de modo que el Operador pueda elegir la que mejor aplique en cada escenario.

- **C003.001.007, RAM**: es necesaria una gran cantidad de memoria intermedia para poder realizar multitud de operaciones matemáticas en el menor tiempo posible.

- **C003.001.008, Alimentación eléctrica**: preveyendo las condiciones adversas de instalación, en las que puede no haber una fuente de electricidad disponible o que ésta falle, se proveerá un sistema autónomo de electricidad.

Nota: al realizarse su interacción desde un equipo remoto, no aplican por tanto ni monitor ni periféricos de entrada como teclado o ratón.

[c003.002] SOFTWARE

- **C003.002.001, Sistema operativo**: es necesario para ejecutar el interfaz un sistema operativo que cumpla las siguientes características, extraídas de los requisitos:

  - GNU: el uso de software libre es solicitado en este proyecto.
  - Fiablidad y estabilidad.
  - Soporte multiprocesador: para facilitar su mantenimiento a lo largo del tiempo, debe ser un sistema operativo con soporte para los distintos tipos de arquitecturas existentes.

- **C003.002.002: Lenguaje de programación para el binario**. Debe ser:
  - Robusto y con control de excepciones.
  - Modular
  - Con gestión de componentes (propiedades, eventos y atributos).
  - Versionable.
  - Multiplataforma
  - Con licencia libre GNU.

- **C003.002.003: Sistema de soporte remoto**. Basado en software libre, debe permitir la conexión a una entidad autorizada. La transmisión de los datos debe llevar un cifrado alto, mínimo de 2048 bits.

- **C003.002.004: Sistema de ficheros**. Debe cumplir las siguientes caracteríticas:
  - Journaling ó bitácora de transcciones.
  - Desfragmentación online.
  - Compatibilidad con la mayoría de sistemas operativos.
  - Alto rendimiento.

- **C003.002.005, Sniffer**: Este componente debe ser, como el resto, de licencia GNU. Las capturas que grabe deben estar en el formato estándar “tcpdump capture v 2.4”.

- **C003.002.006, Decodificador**. Las capturas realizadas deben ser procesadas para eliminar la información de protocolo, dejando solamente la parte de usuario, que es la que se analizará posteriormente. Por eso se necesita un software que procese las capturas de tráfico decodificando al menos los siguientes protocolos que viajen sin cifrar:

PROTOCOLOS NECESARIOS A DECODIFICAR			
HTTP	SMTP	POP3	IMAP
DNS	FTP	SIP	
PROTOCOLOS OPCIONALES A DECODIFICAR			
IPP	PJL	MMSE	TFTP
MSN	GTALK	YAHOO	EMULE
RTP	RTCP	NNTP	SDP
IRC	RTP		

- **C003.002.007, Analizador de datos**
  Este software debe ser capaz de buscar palabras sospechosas en las capturas decodificadas, y ser manejable a través de un interfaz web.

- **C003.002.008, compresor**
  Previo al envío de las capturas al Operador, se comprimirán para disminuir su tiempo de envío. Para ello es necesaria la elección de un compresor, basándose en las siguientes características:
  - Formato de compresión público.
  - Multiprocesador.
  - Multiplataforma.
  - Soporte para recuperación de ficheros compresos corruptos.

- **C003.002.009,** software **de envío de datos**
  Debe cumplir las siguientes características:
  - Envío cifrado.
  - Receptor multiplataforma.

- **C003.002.010: Base de datos**. Este componente debe almacenar los datos manejados en la aplicación de manera organizada. Debe disponer de conectividad segura, transacciones, claves foráneas, triggers, multiplataforma y licencia GNU.

- **C003.002.011: Sistema de impresión**. Este componente debe permitir imprimir a una velocidad habitual. En el propio PC_Sniffer se configurará una impresora de red, componente "**C005.001.001 Impresora"**, para imprimir las alertas. Así se tiene una forma más de que el Operador se entere activamente de que hay una alerta.

- **C003.002.012: Lenguaje de programación para el interfaz**. Debe ser:
  - Modular
  - Con gestión de componentes (propiedades, eventos y atributos).
  - Versionable.
  - Multiplataforma
  - Con licencia libre GNU.

[c004] ServidorDeAlmacenamiento: Máquina en dependencias policiales dedicada a recibir las alertas, el tráfico capturado y configurar el PC_sniffer. Los requerimientos hardware son bajos, hay mucha flexibilidad en ellos, por lo que en este apartado se dará una orientación sobre ellos:

[c004.001] HARDWARE

- **[c004.001.001] Microprocesador**: para el fácil mantenimiento de la aplicación, presentándose una tendencia actual de migración de procesadores de 32 bits a los **64bits**, es necesario elegir estos de cara a evitar un temprano desfase del sistema.

- **C004.001.002 Placa base**: de cara a un fácil mantenimiento del producto, abaratar costes y utilizar tecnología probada y contrastada, es necesario que sea una tecnología extendida y común. Además conviene que sea de un tamaño reducido para su fácil ensamblaje.

- **C004.001.003, Almacenamiento**: necesario para el almacenamiento de una gran cantidad de datos. Debe también soportar condiciones adversas.

Máximo tráfico diario: 648 GB
Requisito R-REN006: almacenar los últimos 7 días
Capacidad necesaria:máx_tráfico_diario*días = 648 GB/día * 7 días = 4536 GB

CONCEPTO	CAPACIDAD
Sistema operativo*	40GB
Ficheros temporales	2 GB
Almacenamiento máximo últimos 7 días	4536 GB
TOTAL	**4578 GB**

* El sistema operativo incluirá interfaz gráfica de usuario, así como una zona de disco dedicada al Operador, para que pueda guardar su información de trabajo.

TOTAL		
Mínimo		4578
Riesgo	10.00%	457'8 GB
TOTAL		**5035'8 GB**

- **C004.001.004, Puerto de conexión auxiliar**: preveyendo futuras ampliaciones o necesidades, se incorporará al menos un puerto auxiliar de conexión **estándar**.

- **C004.001.005, Interfaz de red de entrada**: Interfaz de red para obtener los datos a procesar del usuario monitoreado.

- **C004.001.006, Interfaz de red para conexión con operador**. 3G, ethernet o intranet.

- **C004.001.007, Teclado y ratón**.

[C004.002 SOFTWARE]

- **C004.002.001, Sistema operativo**: es necesario para ejecutar el interfaz un sistema operativo que cumpla las siguientes características, extraídas de los requisitos:
  - GNU: el uso de software libre es solicitado en este proyecto.
  - Interfaz gráfico de usuario: debe disponer de gestor gráfico para facilitar al máximo la interacción del usuario con el sistema.
  - Soporte multiprocesador: para facilitar su mantenimiento a lo largo del tiempo, debe ser un sistema operativo con soporte para los distintos tipos de arquitecturas existentes.
  - Navegador web.

- **C004.002.002: Sistema de impresión**. Este componente imprimirá las alertas e información (como emails) decodificada, etc. Debe ser un producto con interfaz compatible con el interfaz auxiliar **C004.001.004.**

- **C004.002.003: Sistema de ficheros**. Se utilizará el mismo que en el componente **C003.002.004**.

- **C004.002.004: Sistema de interfaz gráfica de usuario**
  Para facilitar el uso del sistema al Operador, y ofrecer una plataforma para la ejecución gráfica de un navegador, es necesario una interfaz gráfica de usuario compatible con X11.

- **C004.002.005: Navegador web.**
  Con la finalidad de interactuar con el sistema, el Operador necesitará un navegador web.

C005.001.001: Impresora. Es necesario disponer de una impresora para obtener las alertas impresas. Debe tener conectividad por IP para ser independiente de cualquier sistema, y compatible con los componentes "**C003.002.011 Servidor de impresión"** y "**C004.002.002** Servidor de impresión"

3.3 TECNOLOGÍAS APLICABLES

Este apartado explica, partiendo de los requisitos anteriormente propuestos, las tecnologías existentes y utilizables para cada caso. Debido a su alta volatilidad, se incluye en el "**Anexo VIII: TECNOLOGÍAS APLICABLES**".

3.4.1 Arquitectura de alto nivel definitiva

Tras el análisis, se propone la siguiente arquitectura:

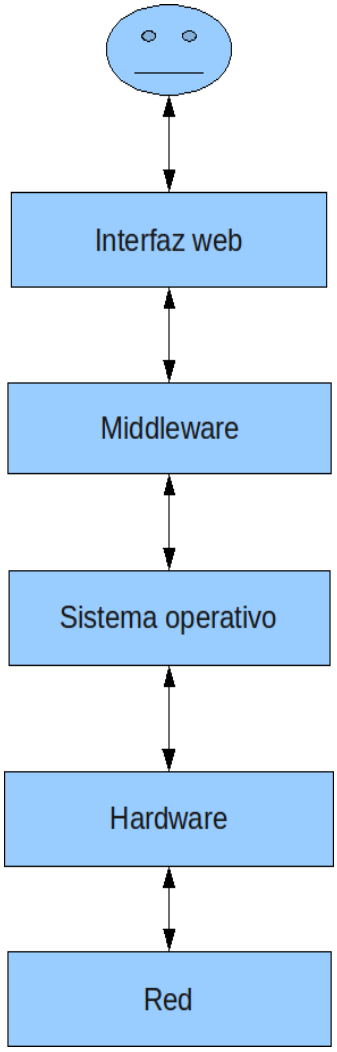
- Red: medio para la obtención de los datos.

- Hardware: electrónica para la obtención de la información.

- Sistema operativo: software para comunicar middleware y hardware.

- Middleware: lógica con la finalidad de procesar la información.

- Interfaz web: GUI para facilitar el uso al operador.



3.4.2 Selección de las tecnologías impuestas

Este apartado explica, partiendo de los requisitos anteriormente propuestos, las tecnologías existentes y utilizables para cada caso. Debido a su alta volatilidad, se incluye en el "**Anexo X: Tecnologías elegidas para cada componente**".

En este sistema pueden ocurrir las siguientes problemáticas:

1	**PROBLEMA**: el usuario monitorizado está lanzando un ataque informático y al llegar al sniffer éste puede sufrir los efectos del ataque.
SOLUCIÓN: Se denegará cualquier paquete (mediante tecnología iptables o similar) proveniente por esa tarjeta para ser procesado por la pila IP, mientras que los paquetes seguirán siendo leídos por la librería pcap.	

2	**PROBLEMA**: el sistema es físicamente detectado y robado por el usuario monitorizado o por un operario del ISP.
SOLUCIÓN: Se blindará por software la máquina para que su disección sea altamente costosa o irrealizable.	

3.5.1 Actualización del análisis de seguridad

Es necesario conocer las publicaciones diarias de seguridad de los siguientes softwares empleados:

a) Mediante la herramienta de Debian de actualización:

```
# apt-get update
# apt-get upgrade
```

Que realizará la actualización automática de las siguientes aplicaciones:

- Debian 5.0
- Tcpdump, tshark.
- Openssh-server
- SCP
- Apache 2.x
- Fail2ban.

b) Mediante los respectivos proveedores:
- Xplico 0.x
- XplicoAlerts

3.6 DISEÑO DEL PLAN DE PRUEBAS

Ver “**Anexo IX: PRUEBAS**”.

Capítulo 4

Diseño arquitectónico

4.1 *USUA*RIOS DEL SISTEMA

Se distinguen dos usuarios del sistema:

- **Usuario monitorizado**: Usuario sospechoso que genera tráfico, el cual el sistema diseñado en el presente proyecto pretende analizar. Este usuario desconoce totalmente el sistema y la actividad de monitorización sobre su tráfico.

- **Operador:** Usuario que controlará el sistema y recibirá las alertas generadas resultado de monitorizar el tráfico del usuario.

4.2 CASOS DE USO

NOMBRE	CU-001
OBJETIVO	Decodificación-y-análisis
ACTOR(ES)	Usuario, Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	El tráfico generado por el usuario queda decodificado y analizado en busca de igualdad completa con ficheros sospechosos susceptibles de alerta. Además el tráfico queda grabado en el servidor de almacenamiento.
ESCENARIO BÁSICO	1. El operador elige los filtros a aplicar al tráfico de un sospechoso y elige “Comenzar”. 2. El sistema comienza la recolección de los datos. 3. El sistema decodifica los datos. 4. El sistema comprueba su similitud con ficheros sospechosos, sin encontrar nada alertante.
ESCENARIO ALTERNATIVO	1. El operador elige los filtros a aplicar al tráfico de unsospecho y elige “Comenzar”. 2. El sistema comienza la recolección de los datos. 3. El sistema decodifica los datos. 4. El sistema comprueba los datos en busca de palabras sospechosas. 5. El sistema comprueba su similitud con ficheros sospechosos, detectando igualdad de ficheros. 5. El sistema genera una alerta.
ERROR	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

CASO DE USO - SIGUIENTE FASE	
NOMBRE	CU-002
OBJETIVO	Configuración
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	La configuración del sistema queda modificada con nuevos parámetros.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de los actuales datos de configuración del sistema. 2. El operador modifica uno o varios de los parámetros de funcionamiento y pulsa “Modificar”. 3. El sistema acepta los cambios y los aplica en caliente.
ESCENARIO ALTERNATIVO	1. El operador solicita la presentación vía web de los actuales datos de configuración del sistema. 2. El operador modifica uno o varios de los parámetros de funcionamiento y pulsa “Modificar”. 3. El sistema, tras validar los datos, no acepta alguno por no ser aplicable (ej: tipo incorrecto de dato) y no aplica **ningún** cambio. 4. El sistema le presenta de nuevo los datos, indicando el error anterior para ser corregido. 5. El usuario reintroduce los datos. 6. El sistema acepta los cambios y los aplica en caliente.
CONDICIONES DE ERROR	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

NOMBRE	CU-003
OBJETIVO	Crear un filtro.
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	Se crea un filtro nuevo y vacío para poder analizar tráficos sospechosos posteriormente con él.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de los actuales filtros y hace click en “Nuevo. 2. El operador inserta el nombre y opcionalmente un fichero con hashes relativos a este crímen. 3. El sistema acepta los cambios y los aplica en caliente.
ESCENARIO ALTERNATIVO	3. El sistema, tras validar los datos, no acepta alguno por no ser aplicable (ej: tipo incorrecto de dato) y no aplica ningún cambio. 4. El sistema le presenta de nuevo los datos, indicando el error anterior para ser corregido. 5. El usuario reintroduce los datos. 6. El sistema acepta los cambios y los aplica en caliente.
CONDICIONES DE ERROR	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

NOMBRE	CU-004
OBJETIVO	Eliminar un filtro.
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	El filtro elegido así como sus elementos asociados quedan eliminado del sistema.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de los actuales filtros y hace click en “Eliminar” en el icono correspondiente del filtro a borrar. 2. El sistema elimina el filtro y muestra de nuevo la lista actualizada.
ESCENARIO ALTERNATIVO	1. El operador solicita la presentación vía web de los actuales filtros y hace click en “Eliminar” en el icono correspondiente del filtro a borrar. 2. El sistema está actualmente en ejecución con ese filtro seleccionado. 3. El sistema se detiene, avisándolo.
CONDICIONES DE ERROR	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

CASO DE USO - SIGUIENTE FASE	
NOMBRE	CU-005
OBJETIVO	Añadir una palabra a un filtro.
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	Se añade una palabra a un filtro.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de los actuales filtros y hace click en uno de ellos. 2. El sistema muestra un formulario para añadir una palabra el filtro. 3. El usuario rellena los campos y hace click en “Enviar”. 4. El sistema valida los datos, incorpora la palabra al filtro y muestra un mensaje con el resultado positivo de la operación.
CONDICIONES DE ERROR	4. El sistema valida los datos encontrando alguna incorrección, y muestra un mensaje de error. 5. El sistema muestra un formulario para añadir una palabra el filtro. 6. El usuario rellena los campos y hace click en “Enviar. 7. El sistema valida los datos, incorpora la palabra al filtro y muestra un mensaje con el resultado positivo de la operación.
NOMBRE	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

CASO DE USO - SIGUIENTE FASE	
NOMBRE	CU-006
OBJETIVO	Eliminar una palabra de un filtro
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	Se elimina una palabra a un filtro.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de los actuales filtros y hace click en uno de ellos. 2. El sistema muestra la lista de palabras existentes. 3. El usuario elige la opción “Eliminar” de una de ellas. 4. El sistema elimina la palabra e informa con mensaje con el resultado positivo de la operación.
CONDICIONES DE ERROR	N/A
NOMBRE	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

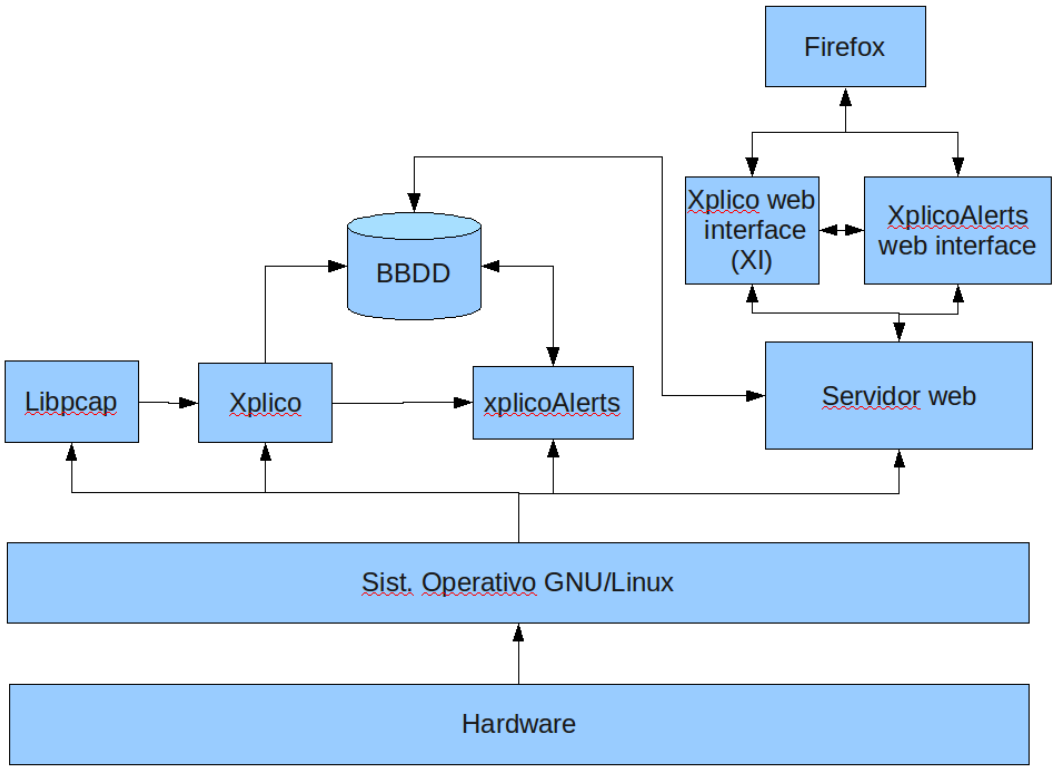
NOMBRE	CU-007
OBJETIVO	Consultar alertas
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	Se muestra por pantalla una alerta.
ESCENARIO BÁSICO	1. El operador solicita la presentación vía web de la lista de alertas. 2. El operador selecciona una alerta en concreto. 3. El sistema muestra la información de la alerta y la propia información que ha generado la alerta.
CONDICIONES DE ERROR	N/A
NOMBRE	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

CASO DE USO - SIGUIENTE FASE	
NOMBRE	CU-008
OBJETIVO	Descargar capturas
ACTOR(ES)	Operador
PRECONDICIONES	No aplica.
POSTCONDICIONES	Se descarga en local la captura de tráfico.
ESCENARIO BÁSICO	1. El Operador solicita la presentación vía web de la lista de capturas. 2. El operador selecciona una captura en concreto. 3. El sistema descarga la captura en el equipo del Operador.
CONDICIONES DE ERROR	N/A
NOMBRE	Si se produce cualquier error, el sistema mostrará un mensaje y finalizará la interacción del escenario (sin éxito).

Capítulo 5

Diseño detallado

Se establece el siguiente esquema de diseño:



El funcionamiento general del sistema consistirá en:

1º) Obtención del tráfico de red: El software Xplico leerá de la red las tramas de datos mediante la librería Libpcap.

2º) Decodificación: el software Xplico grabará en disco duro y en la base de datos la información obtenida.

3º) Evaluación de la información: El plugin “xplicoAlerts” evaluará toda la información decodificada cotejándola con los patrones existentes de tráfico sospechoso almacenado en la base de datos.

4º) Interfaces web: el usuario podrá acceder tanto a la información decodificada como a las alertas generadas a través de sendos interfaces web.

5.1.1 INTERCEPTOR

Elemento capaz de copiar una señal física en otra línea sin alterar la primera.

EL DISEÑO HARDWARE DE ESTE ELEMENTO NO APLICA EN ESTE DOCUMENTO

5.1.2 MODEM ADSL USB / MINIPCI

Modificación del firmware de un router para convertir señales ADSL en IP, redireccionándolas a un canal concreto.

EL DISEÑO HARDWARE DE ESTE ELEMENTO NO APLICA EN ESTE DOCUMENTO

5.2.1 Diseño del componente “Sniffer”

Descripción teórica del componente, susceptible a sucesivas revisiones.

Se arrancará el sniffer para que inicie la grabación de la captura de tráfico en ficheros de captura. Estos se pasarán posteriormente al segundo paso del flujo, el procesado de Xplico. Los ficheros se podrían grabar de dos maneras distintas:

- Por tiempo: cada n-tiempo (1h) se cambia el fichero en el que se guarda el tráfico. Éste se nombraría de la forma “*traffic.capture_SOSPECHOSO_DATE_HOUR.pcap*”.

- Por tamaño: alcanzados los n-megas, se pararía de grabar en ese fichero y se continuaría, sin pérdida de datos, en otro. Se presenta el problema de que si un sospechoso genera poco tráfico, el tratamiento de su información puede demorarse o incluso no llegar a hacerse nunca (en el caso de no alcanzarse el mínimo de tráfico para pasarlo a la siguiente fase).

Problemáticas existentes:

- tcpdump escribe los datos con cierto retraso, por lo que es necesario encontrar la manera de que escriba lo más cercano a tiempo real. Se puede estudiar el valor de n-segundos que utiliza como buffer.

- Si se le pasasen a Xplico capturas fragmentadas, se podría perder información, al enviarse por ejemplo una imagen sospechosa que procesada por separado como dos sub-imágenes no generase alarma.

Pruebas:

Creación de dos tuberías para tener acceso a la información y redireccionarla posteriormente a Xplico.

```
$ mkfifo capture.cap
$ mkfifo Xplico.input.data.cap
$ tcpdump -i eth0 -C 10 -ns 1514 -w /tmp/captures/capture.cap -ttt
$ tail -f capture.cap >> Xplico.input.data.cap
$ sudo xplico -f pcap -f Xplico.input.data.cap
```

Es necesario diseñar un sistema para que la misma información que hay en la tubería vaya a un fichero PCAP, en tramos de una hora y sin perder paquetes en el proceso de cambio de fichero en la hora siguiente. Opciones a estudiar:

A) De una captura, copiar sólo los datos pertenecientes a un rango.

```
$ mkfifo capture.cap
$ tcpdump -i eth0 -C 10 -ns 1514 -w /tmp/captura.pcap
$ tail -f captura.pcap |./filter-time.pl '17:00:00' 22:00:00'
```

```
#!/usr/bin/perl
#Please note, that tcpdump's output didn't contain day part, so you
should care about time ranges over midnight like 23:00 -> 01:00.
# Convert time range to seconds from midnight
$ARGV[0] =~ /(\d{2}):(\d{2}):(\d{2})/;
$begin_t = $3 + 60 * $2 + 60 * 60 * $1;
$ARGV[1] =~ /(\d{2}):(\d{2}):(\d{2})/;
$end_t = $3 + 60 * $2 + 60 * 60 * $1;

while (<STDIN>) {
 # If row begin with time stamp
 if (/^(\d{2}):(\d{2}):(\d{2}).\d{6}/) {
   $time = $3 + 60 * $2 + 60 * 60 * $1;
   # We make decision is it packet in range or not
   if ($time <= $end_t && $time >= $begin_t) {
    $print_packet=1
   } else {
    $print_packet=0
   }
 }

 if ($print_packet) { print $_ }
}
#http://www.experts-
exchange.com/Networking/Unix_Networking/Q_24308158.html
```

B) Wireshark ofrece un comando para filtrar datos por fecha/hora de una captura. Se puede estudiar su ejecución concurrente con la de otros sniffers.

```
$ wireshark -r input.tcpdump -w output.tcpdump -R 'frame.time >=
"Aug 1, 2001 00:52:34" && frame.time <= "Aug 1, 2001 00:55:34"
```

C) Ejecutar a las horas en punto una instancia de sniffer que durará una hora exacta. Esta solución no ofrece precisión, ya que muchos paquetes del primer segundo se podrían perder al no haber arrancado totalmente el sniffer correspondiente.

Code:

```
while [ 1 ];
  do
  DATE=`date '+%m-%d-%Y-%a:%H:%M'`;
  /usr/bin/timed-run 3600 /usr/sbin/tcpdump -nei eth0 -s 1515 -w \
  /path/to/dump.$DATE.lpc;
    /usr/bin/find /path/to/ 'dump.*.lpc' -mtime +.04 -exec /usr/bin/rm
{} \;
  done
```

D) "Wireshark -d": ejecuta el sniffer Wireshark en modo "múltiples ficheros". Se le especifica una duración (3600 segundos) y al cabo de ese tiempo comenzará a escribir en un nuevo fichero. No obstante no cubre directamente la necesidad de conservar los paquetes de una determinada hora en un fichero (ej: inicio de la aplicación a las 19:37)

```
$ tshark  -a  captura20090930_0100_0159.pcap duration:3600
'frame.time >= "Aug 1, 2001 00:52:34" && frame.time <= "Aug 1,
2001 00:55:34"
```

Aplicación práctica #1
Ejecución de dos sniffers con esta opción, durando el primero los segundos que queden hasta la siguiente hora en punto, y empezando el segundo sniffer en esa *hora en punto*, el cual además generaría un nuevo fichero cada esos n-segundos (n=3600+margen_de_error)

```
$  tshark  -d [segundos hasta la 1:00am + por_si_acaso] -i eth0
-w captura_000000 -R 'frame.time >= "Aug 1, 2001 00:34:34"
&& frame.time <= "Aug 1, 2001 00:59:59"    &

$  wireshark –b duration:3600  eth0 -w captura.cap -R
'frame.time >= "Aug 1, 2001 01:00:00" && frame.time <= "Aug
1, 2001 01:59:59"

$(date +%Y%M%d-%H%M%s).pcap
```

Aplicación práctica #2
Realizar las capturas por días. Pese a que se procesará en tiempo real, el acceso de las capturas de datos sería en diferido. Se ejecutaría un sniffer antes de iniciarse un día (a las 23:58) y se finalizaría a las 0:05 del día siguiente al deseado, por prevenir que debido a que la CPU esté muy saturada y se demoren ciertas acciones. Es una adaptación de la "**Aplicación práctica #1**"

5.2.2 Diseño del componente "Decodificador"

Decodificador de las capturas, extrayendo su información en modo legible, de modo que se pueda procesar a nivel INDECT-WP3 e INDECT-WP4. Se utilizará el software **Xplico**. No cubre todos los protocolos estandarizados, pero si los requeridos por los requisitos, a parte de otros, y se está desarrollando continuamente soporte para otros protocolos.

Recibe las capturas mediante una tubería:

```
$ ./sudo xplico -m pcap -f tuberia.pcap
```

Y decodifica la captura, generando dos salidas distintas:

a) **./tmp/xplico/[protocolo]**: tráfico decodificado, en el que se separan la parte de protocolo y la parte de usuario. Ej:

```
$ wget terra.es
$ ls tmp/xplico/http
(…)
$file tmp/xplico/http
(…)
```

b) **Captura de tráfico**: flujos no decodificados, para procesar manualmente si se desea.

```
$ telnet cisco.com 8464
$ ls tmp/xplico/
(…)
$file tmp/xplico/(…)
(…)
```

5.2.3 Diseño del componente “xplicoAlerts”

Este componente es el núcleo del sistema propuesto. Coordina todas las aplicaciones y el flujo de la información. Dispone también de una parte con interfaz web para que el Operador pueda interactuar con la aplicación. Estará compuesto de:

- **Binario:** Código C++ con toda la lógica del programa.
  - Conocer el estado del sistema: saber si está en funcionamiento o no y desde cuando.
  - Arrancar/Parar el sistema y elección de los filtros a aplicar al usuario.
  - Opciones de configuración: cambiar variables de configuración.
  - Ver log del sistema.
  - Sistema de envío: para el caso en el que el canal entre el sniffer y el Operador sea reducido, tendré que deshabilitar el envío automático de ficheros, pero ofrecer al Operador se los baje mejor él por la web.
  - Rotación de logs y ficheros decodificados.
  - Dar de alta / eliminar filtros.
  - Dar de alta / Eliminar una palabra de un filtro.

- **Interfaz web:** Web para facilitar al Operador la interacción con el sistema. Se comunicará con el binario a través de una *tubería* o *pipe*.
  - Visualizar alertas.
  - Descargar capturas.
    Automonitorización: web de Monit.

5.2.3.1 Tablas de almacenamiento de datos.

TABLA 1	fileshashes	
Datos de los ficheros sospechosos		
Campo	**Formato**	**Significado**
id	INTEGER PRIMARY KEY	Identificador único del hash.
file	varchar(256)	(opcional) Nombre del fichero.
Hash	varchar(256)	Hash del fichero.
crime_id (cambiar por category_id)	TEXT	(foreign key → crimes:id) Crimen en el que este hash se debe considerar sospechoso.

TABLA 2	categories	
Categorías o crímenes existentes.		
Campo	**Formato**	**Significado**
id	INTEGER PRIMARY KEY	Identificador único de la categoría.
crime	TEXT	Nombre de la categoría o crímen.

TABLA 3	alerts	
Alertas generadas durante el procesamiento de la información.		
Campo	**Formato**	**Significado**
id	INTEGER PRIMARY KEY	Identificador único de la alerta.
date	TEXT	Fecha y hora del paquete de datos con la palabra sospechosa.
tittle	TEXT	Título de la alerta
message	TEXT	Contenido descriptivo de la alerta
notes	TEXT	Notas del operador.
filePath	TEXT	Ruta del fichero decodificado que ha generado la alarma.

A través del campo filePath se presenta un enlace a Xplico para visualizar la información que ha generado la alerta, a parte de ofertar la descarga de la captura de tráfico.

asdsadwqq

TABLA 4 - Tabla palabras sospechosas		
Palabras sospechosas		
Campo	**Tipo**	**Significado**
Word	varchar(256)	Palabra sospechosa en una temática
Category	NUMBER	(foreign key → crimes:id) Crimen en el que este hash se debe considerar sospechoso.
Value	INTEGER	(futuros usos) Peso de esa palabra en ese crímen para generar una alerta.

TABLA 5 - Tabla URL's sospechosas		
Palabras sospechosas		
Campo	**Tipo**	**Significado**
URL	varchar(256)	Palabra sospechosa en una temática
Category	NUMBER	(foreign key → crimes:id) Crimen en el que este hash se debe considerar sospechoso.
Value	INTEGER	(futuros usos) Peso de esa URL en ese crímen para generar una alerta.

5.2.3.2 Ficheros

Tanto en el equipo del sniffer como en el del Operador toda actividad que se realice quedará reflejada y sincronizada en los siguientes ficheros de log.

Fichero log de Sistema_de_envíos log.files.sent.DATE.log		
Registro de todas las actividades realizadas por este componente. Información replicada en ambos equipos.		
Campo	**Formato**	**Significado**
Fecha	dd/mm/aaaa	Fecha fin del envío
Hora	Hh:mm:ss	Hora fin del envío
IP destino	Xxx.xxx.xxx.xxx	IP del Operador
Puerto destino	xxxxx	Puerto del Operador.
Fichero enviado	String[512]	Alerta o captura comprimida.
Path	String[512]	Ruta destino

Fichero log de CASO log.alerts.$DATE.log		
Registro de todas las actividades realizadas por este componente. Información replicada en ambos equipos.		
Campo	**Tipo**	**Significado**
Fecha_alerta	dd/mm/aaaa	Fecha de generación de la alerta.
Hora_alerta	Hh:mm:ss	Hora de generación de la alerta.
Fecha_captura	dd/mm/aaaa	Fecha del paquete de datos causante de la alerta.
Hora_captura	Hh:mm:ss	Hora del paquete de datos causante de la alerta.
Palabra_sospechosa	String[512]	Palabra causante de la alerta.
Contexto	String[100]	50 bytes anteriores y 50 posteriores.

Se ubicarán en **/opt/xplico/xplicoAlerts/logs** y se irán rotando con la periodicidad especificada en su variable **rotateLogsEveryNDays**.

"alertsSystem" utilizará un fichero de configuración con los siguientes parámetros:

DATOS NECESARIOS		
COMPONENTE	**DATO**	**VALOR POR DEFECTO**
Xplico	Xplico_WebServer_IP	
Xplico	Xplico_WebServer_Port	
Xplico	Xplico_WebServer_Username	
Xplico	Xplico_WebServer_Password	
Xplico	Xplico_LOG_DIR_PATH	
Xplico	Xplico_lastdata.txt_file	
Xplico	Xplico_TMP_DIR_PATH	
Xplico	Xplico_Daemon	/etc/init.d/xplico
xplicoAlerts	Local_interface_to_sniff	eth0
xplicoAlerts	Local_interface_for_backup	eth2
xplicoAlerts	Core_WebServer_IP	
xplicoAlerts	Core_WebServer_Port	
xplicoAlerts	CORE_Daemon	/etc/init.d/xplicoAlerts
xplicoAlerts	rotateCapturesEachNDays	7
xplicoAlerts	rotateLogsEachNDays	7
xplicoAlerts	Pipe_path	/tmp/xplicoAlerts.pipe
xplicoAlerts	sniffer_path	/usr/bin/tshark
xplicoAlerts	Captures_path	/home/indect/captures
xplicoAlerts	pathDataBase	/opt/xplico/xplico.db

xplicoAlerts	Printer	http://172.26.0.8:9100
Sistema de envío	Compressor	pigz
Sistema de envío	SCP_Path	/usr/bin/scp
Sistema de envío	Local_interface_to_Operator	eth1
Sistema de envío	Destination_IP	
Sistema de envío	Destination_port	22
Sistema de envío	Clave pública PC_sniffer	
Sistema de envío	Clave pública PC_Almacén	
Sistema de envío	Clave privada PC_sniffer	
Sistema de envío	Clave privada PC_Almacén	
Sistema de envío	signature	
Sistema de envío	Remote_Path_to_store	/home/snipol/captures
Sistema de envío	Autosend_captures_to_Operator	true
Monit	Monit_IP	
Monit	Monit_Port	
Monit	Monit_Username	
Monit	Monit_Password	
Monit	Monit_Service	/etc/init.d/monit
Monit	Alert_HD_busyPercent	70
Monit	Alert_RAM_busyPercent	70
Monit	Alert_PING_lost	3
Monit	Alert_CPU_busyPercent	70
Monit	Alert_TEMP_higherThan	40

Así mismo el sistema de monitorización utilizará un fichero propio ubicado en **/etc/monit/monitrc**

5.2.3.3 Elementos

Este software se divide en 3 elementos:

5.2.3.3.1 Analizador de ficheros.

Este elemento busca ficheros decodificados y los compara con una base de datos, buscando semejanza para poder dictaminar una alarma.

Las funciones que contienen la lógica de este software son:

- trainingFilters();
- hash(word)
- isWordInFilter(word,filter)

Las cuales, para explicarlas, requieren antes conocer el uso de **Bloom Filters**.

Ante la ingente cantidad de datos que se prevé manejar en ese sistema, y la velocidad de procesamiento exigida (**R-REN001**), es necesario utilizar un sistema de alto rendimiento para evitar una serie continuada de consultas a una BD.

Por ello se plantea el uso de **Bloom filters**. La finalidad de estos es comprobar si una palabra pertenece a un conjunto de ellas. La mecánica resumidamente consiste en codificar una serie de palabras en un mapa de bits, inicialmente a cero, mediante unas funciones hash. Estas funciones hash indican qué bits del mapa hay que activar. Posteriormente, si se quiere comprobar si una palabra cualquiera pertenece a un conjunto, se procesa por las mismas funciones hash y se comprueban todas si las posiciones del mapa de bits que indican las funciones hash en esta palabra han sido marcadas durante el entrenamiento. Si ocurre el caso, la palabra pertenece al conjunto, de otra manera, con que un sólo bit no esté marcado, no pertenecerá al conjunto.

Ejemplo:

Mapa de bits inicial:

Posición	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Valor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Entrenamiento:

"bomba"

Hash1(bomba) = 4
Hash2(bomba) = 3

"antrax"

Hash1(antrax) = 2
Hash2(antrax) = 6

"dinamita"

Hash1(dinamita) = 4
Hash2(dinamita) = 17

Por lo que tras este sencillo entrenamiento habría que marcar a "1" las posiciones 4, 3, 2, 6, 4 y 17 del mapa de bits. Nótese que la posición 4ª está repetida, es un caso habitual debido a que en las funciones hash pueden repetirse resultados. Por tanto, el mapa de bits quedaría del siguiente modo:

Mapa de bits entrenado:

Posición	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Valor	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0

A partir de aquí se podría comenzar a probar la pertenencia o no de palabras al conjunto:

"pared"

Hash1(pared) = 11
Hash2(pared) = 14

La cadena "pared" consultaría las posiciones 11 y 14 del array. Al ser al menos una de ellas "0", se deduciría que **no pertenece** al conjunto de palabras inicial.

"antrax"

Hash1(antrax) = 2
Hash2(antrax) = 6

La cadena "antrax" consultaría las posiciones 2 y 6 del array. Al ser ambas "1" , se concluiría que **pertenece** al conjunto de palabras inicial.

"asfalto"

Hash1(antrax) = 2
Hash2(antrax) = 17

La cadena "asfalto" consultaría las posiciones 2 y 17 del array. Al ser todas "1" , se concluiría que pertenece al conjunto de palabras inicial. Esto es un ejemplo de "**falso positivo**".

La naturaleza de los Bloom Filters por tanto posibilita la existencia de **falsos positivos** pero nunca la de falsos negativos. Esto facilita la labor de observación y vigilancia de contenidos sospechosos en este proyecto. Así mismo, se modificará el código para que antes de dar un falso positivo, se corrobore mediante una consulta directa a la base de datos, la pertenencia real de esa palabra al filtro.

Además, las hashes internamente **se almacenarán en mayúsculas**, y se evaluarán así también, para no forzar al Operador a introducir toda la combinatoria posible.

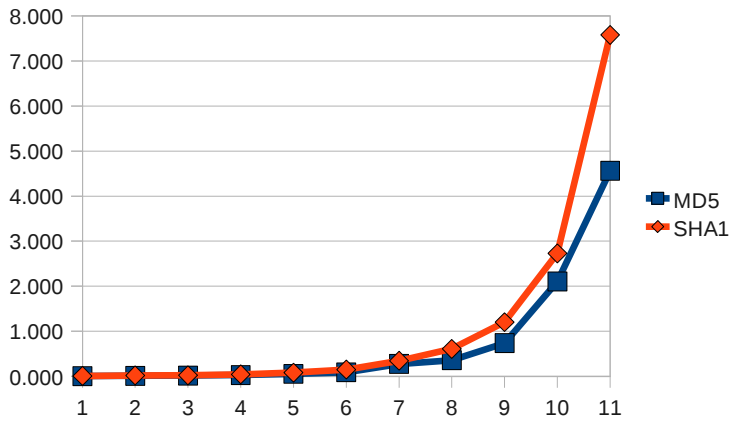
– **Algoritmo para generación de hash**:

Una función de hash es una función para resumir o identificar probabilísticamente un gran conjunto de información, dando como resultado un conjunto imagen finito de menor tamaño (un subconjunto de los números naturales por ejemplo). Varían en los conjuntos de partida y de llegada y en cómo afectan a la salida similitudes o patrones de la entrada. Una propiedad fundamental del hashing es que si dos resultados de una misma función son diferentes, entonces las dos entradas que generaron dichos resultados también lo son.

COMPARATIVA		
Característica	**MD5**	**SHA1**
Longitud (bits) de la salida	128	160
Dificultad para conseguir dos mensajes con igual clave (na operaciones)	264	280
Tamaño bloque procesado (bits)	512	512
Nº de funciones primitivas	4	4
Nº de operaciones	64	80
Bits de buffer a procesar	128	160
Claves calculadas por unidad de tiempo	3	1
Constantes utilizadas	64	4

Tiempo (segundos) empleado en hashear un fichero.

Tamaño (MB)	MD5 (seg)	SHA1 (seg)
1	0.01	0.01
2	0.01	0.02
4	0.02	0.02
8	0.03	0.04
16	0.06	0.08
32	0.09	0.15
64	0.28	0.35
128	0.36	0.61
256	0.74	1.2
512	2.1	2.73
1024	4.56	7.58



Durante la implementación del proyecto se ha descubierto el algoritmo **Fuzzy hashing** en sus implementaciones "**ssdeep**" y su variante "**DeepToad**", que generan hashes relacionables. Consiste en una técnica utilizada para generar 'hashes' que, a diferencia de los hashes criptográficos donde un cambio en un byte provoca que todo el resumen generado sea totalmente diferente (esto es, un cambio es propagado al resultado completo final) en este caso el resumen general no cambia o bien varía muy ligeramente. Es decir, no se busca que los resúmenes generados estén libres de colisiones sino más bien lo contrario: dos archivos prácticamente iguales han de ser clasificados como iguales o muy semejantes.

Se realizará una re ingeniería para aplicar este algoritmo sustituyendo al implementado sha1, lo cual aumentará la posibilidad de identificar ficheros iguales o alterados ligeramente para evitar su detección en los filtros.

5.2.3.3.2 Analizador de cadenas.

Este software procesará una serie de ficheros buscando cadenas sospechosas. Si encuentra una, generará una alerta, la enviará y continuará la búsqueda. Los ficheros decodificados no se borrarán para poder ser consultados por el interfaz web posteriormente, por ello se realizará una lista de ficheros ya procesados.

El funcionamiento, a nivel de pseudocódigo de alto nivel, será el siguiente:

```
ArrancarSistema()

//Educación del filtro.
semaforoEstaCasoListo = false
conjuntoPalabras=leerTodasPalabrasBBDD(filtro)
porCadaPalabra (conjuntoPalabras)
    educarBloomFilter(palabraAlerta)
semaforoEstaCasoListo = true

//Actividad normal
mientras (1) //u otro parámetro de control.
        while (exists a not processed file)
                buscarPalabra()
                hashear(palabraSospechosa)
                buscarEnBloomFilter(hashPalabraSospechosa)
                si esSospechosa entonces
                        verificarDirectamenteEnBBDD( palabraSospechosa)
                                si esSospechosaVerificado entonces
                                        generarAlerta(word, filter)
                                si no
                                        falsoPositivo, loggearlo
```

Cabe destacar que se producirá un alto consumo al arranque del programa,en el momento de educación del filtro, pero que favorecerá enormemente el rendimiento del software en el resto de la ejecución al poder procesar palabras de una forma más sencilla y menos costosa en términos de CPU que realizando una consulta a una base de datos por cada una de las cientos de palabras que viajen en un paquete IP.

La base de datos contendrá las palabras con las que entrenar cada Bloom filter. Es necesaria ya que el operador, dependiendo del perfil de cada sospechoso, podrá añadir o quitar ciertas palabras.

Además, las palabras internamente **se almacenarán en mayúsculas**, y se evaluarán así también, para no forzar al Operador a introducir toda la combinatoria posible; así mismo, se normalizarán las palabras eliminando las tildes y diéresis tanto en los filtros como en las capturas de tráfico, preveyendo faltas ortográficas del sospechoso, intencionadas o no, que podrían despistar el sistema, por lo que se incluirán las permutaciones de las palabras (“bomba” y “~~vomba~~”). Para este caso, se plantearán los siguientes conjuntos reducidos para entrenar los filtros de detección de palabras sospechosas en una línea ADSL.

EJEMPLOS DE CONJUNTOS DE ENTRENAMIENTO			
TERRORISMO	**NARCOTRÁFICO**	**PEDRASTIA**	**CRACKING**
cócteles	*Droga*	*pedofilia*	*exploit*
cocteles	*gramos*	*pedo*	*man*
molotov	*Cocaína*	*menor*	*middle*
temporizador	*coca*	*lolita*	*attach*
Bomba	*planeadoras*	*teen*	*ddos*
Bomba	*motora*	*pubertad*	*firewall*
lapa	*armas*	*intercambio*	*brute*
trampa	*báscula*	*infantil*	*force*
granada	*hachís*	*niño*	*spoofing*
granadas	*punto*	*niña*	*proxy*
Granada	*entrada*	*desnudo*	*crack*
Granadas	*estrecho*	*desnuda*	*cracking*
pistola	*Gibraltar*	*bebé*	*hack*
armas	*Galicia*	*baby*	*hacking*
antrax	*Colombia*	*cuidadora*	*telnet*
dinamita	*piedra*	*canguro*	*puerto*

Recordar que aunque alguna de estas palabras parezca de uso general o se prevea que su frecuencia va a ser habitual, al buscarse en el contenido del tráfico de un sospechoso concreto cobra mucho más valor.

Inicialmente las funciones a utilizar son:

- **SHA1(word)**: Función que dada una palabra, indicará que bits hay que marcar o consultar en el mapa de bits.
- **TrainingFilters()**; Función que realizará el entrenamiento del mapa de bits del bloom filter. Leerá todas las palabras de una base de datos e irá marcando a “1” los bits que cada palabra indique en el mapa de bits.
- **isWordInFilter(word,filter)**: función encargada de consultar si una palabra, mediante las funciones hash, pertenece al conjuntod de palabras buscadas.
- alert(palabra): generará un aviso, grabándolo en:
  - Log local.
  - Log remoto (operador)
  - Base de datos (visualizable por el interfaz web)
  - Impresora IPP: es necesario alertar al operador de manera **activa**.

Se empleará la utilidad de motores de búsqueda opensource **Lucene** este desarrollo.

Por desarrollar en posteriores fases de este proyecto.

5.2.3.3.3 Analizador de URLs.

Se diseñará un componente para cotejar los intentos de conexión tanto por IP como por DNS, generando alertas consecuentemente.

Por desarrollar en posteriores fases de este proyecto

5.2.4 Diseño del componente “Sistema de envío”

La información capturada se replicará al puesto del operador. Para ello mediante un canal cifrado SSH se enviará compresa mediante la utilidad ***scp***.

Se enviará:

- Captura de tráfico en bruto cada hora: thread dedicado al efecto. Toma una captura de una hora ya pasada, la comprime y la envía. Se ejecutará cada hora.
- Alertas: en tiempo real.

Utilidad de envío: “scp”, Secure Copy. Parámetros:

-
- “-2”: Uso del protocolo 2 en el envío.
- “-B”: Modo “batch”, previene de peticiones de passwords.
- “-l limite”: Limita el ancho de banda, especificado en Kb/s, de modo que no se utilice el canal entero para el envío de un fichero, permitiendo la posibilidad de enviar alertas en tiempo real.
- “-c”: cifrado alto.
- “-C”: comprimir. No se utilizará al haber sido compresa la información anteriormente con un algoritmo de compresión superior al de SCP.

Las capturas residirán en /home/indect/captures. El Sistema de envío realizará la siguiente rutina:

1º) Eligirá el fichero más antiguo de /home/indect/captures en base a su nombre.
2º) Lo firmará.
2º) Lo comprimirá.
3º) Lo enviará al PCOperador por SCP.
4º) Lo moverá a /home/indect/captures/sent.captures para no vovler a ser procesado.

5.2.5 Diseño del componente “Automonitorización”

Este componente realiza labores de autodiagnóstico del sistema para prever situaciones problemáticas:

CHEQUEOS		
Elemento monitorizado	**Condición**	**Acción**
CPU	>70%	Alerta
CPU	>95%	Alerta y Reboot
RAM en uso	>70%	Alerta
RAM en uso	>95%	Alerta y Reboot
Espacio libre en disco	>70%	Alerta
Espacio libre en disco	>85%	LiberarEspacio() Alerta
Espacio libre en disco	>95%	Alerta y Reboot
Temperatura	>40ºC	Alerta
Temperatura	>45ºC	Alerta y Reboot
Ping Operador	>= 10 echos timeout	Ifdown eth1; ifup eth1 Alerta
Ping Operador	>= 100 echos timeout	/etc/init.d/networking restart Alerta
Ping Operador	>= 1000 echos timeout	Alerta y Reboot
Eth0	Received bytes == yesterday Received bytes	/etc/init.d/networking restart Alerta
BBDD	Conect to BBDD:port failed	/etc/init.d/postgresql restart Alerta
Xplico	Conect to WebServer:port failed	/etc/init.d/xplico restart Alerta
Xplico	Pid not found	/etc/init.d/xplico restart Alerta
Core	Pid not found	/etc/init.d/snipol restart Alerta
SAI	SAI_energy <95%	Alerta
SAI	SAI_energy <20%	SAI_energy <95%
SAI	SAI_energy <2%	Halt

Estos umbrales serán editables vía web en la propia página de Sistema.

Para el desarrollo de este componente de utilizará **Monit**, aplicación software libre ampliamente utilizada para monitorizar sistemas y redes.

La integración web se facilitará con el siguiente adaptador:
http://mmonit.com/monit/dist/contrib/monit.php.txt

Así mismo, para algunas partes será necesario crear rutinas nuevas para este caso:

- Fichero de configuración de Monit.
- Rutina de eliminación de ficheros más antiguos: rutina que dado un path, elimina el fichero más antiguo. Consecuentemente debe borrar también la información en la base de datos de Xplico para que las referencias a esos ficheros no permanezca y por tanto sea accesible desde el interfaz web.
- Rutina para liberar CPU: Si el uso de CPU está entre 80 y 95%, ejecutar una rutina para evitar colapso del sistema.
- Servicio de arranque/reinicio/parada de Xplico.

Las alertas, al no disponerse de conexión con Internet para su propio uso, se generarán mediante la impresión en papel de las mismas.

5.2.5.1 Fichero de configuración de Monit

Integración de los siguientes pasos en el software:

1º) Copiar las variables del fichero de configuración que correspondan en una plantilla predefinida de Monit.

2º) Copiar ese fichero a /etc/monit/monitrc

```
$ cp NEW.monitrc /etc/init.d/monit restart
```

3º) Reiniciar el servicio:

```
$ /etc/init.d/monit restart
```

5.2.5.2 Rutina de eliminación de los ficheros más antiguos

Rutina para eliminar el fichero más antiguo de un directorio. Monit la ejecutará tantas veces como sea necesario, hasta que tenga espacio libre y por tanto no se cumpla la condición que la invoca.

```
#!/bin/bash
ls -t1 $1 | tail -1  | xargs  rm
```

Los directorios donde se debe ejecutar esta rutina se definirán en sucesivas revisiones de este documento.

5.3.1 Estructura de directorios

El plugin desarrollado vendrá incorporado en el paquete de Xplico, no así las bases de datos de material e información sospechosa.

/opt/xplico/bin/alertsSystem
/opt/xplico/xplico.db
/opt/xplico/xi/app [components] [models] [views]

En posteriores revisiones de este documento se incorporarán:

- DIR_BACKUP
  - Backup diaria de la BBDD.
  - config.core.xml.back
- Sistema_envio
  - DIR_ captures
- DIR_captures.sent
- logs
  - Logs de cada día.

5.3.2 Servicio de arranque/rearranque/parada de Xplico.

Para poder manejar el software Xplico, se ha creado un servicio (también denominado demonio en la terminología Linux). Su uso es el siguiente:

ÓRDEN	DESCRIPCIÓN
$ /etc/init.d/xplico start	Arranque de la aplicación
$ /etc/init.d/xplico restart	Parada y rearranque de la aplicación
$ /etc/init.d/xplico stop	Para de la aplicación

5.3.3 Servicio de arranque/rearranque/parada de xplicoAlerts

Se implementará un servicio o demonio semejante al de Xplico para xplicoAlerts.

5.3.4 Blindaje del sistema

Al manejarse información confidencial, se tomarán las siguientes medidas de cara a garantizar la seguridad de la misma:

- **Acceso local deshabilitado**. Sólo se permitirá acceder por conexión cifrada.
- **Conexiones de red**: se calculan tres interfaces de red con políticas de filtrado y seguridad de las mismas:
  - Ignorar todo paquete que venga por la interfaz de escucha pero procesarlo con el sniffer.
  - Ignorar todo paquete que venga por la interfaz de Operador excepto los provenientes de la IP_Operador.
- **Acceso remoto:** permitido sólo por SSH desde una IP concreta, con un canal cifrado y con una clave pública concreta, además de password. Para evitar ataques por fuerza bruta, se configurará el sistema para que después de 5 intentos de conexión fallidos, se restrinja su acceso durante 20 minutos mediante el software *fail2ban*.
- **Cifrado de disco**: se cifrará el disco con para evitar su desmontado y ensamblado en otra máquina, tal y como se ha visto en el apartado correspondiente, con **LUKS**.

Las configuraciones y detalles técnicos se incluirán en un anexo en posteriores versiones.

5.3.5 Diseño componente “PCAlmacenamiento”

Este componente está orientado a recibir las capturas de tráfico compresas que le envía el PCSniifer. Para ello dispondrá de un servidor ssh, tipicamente OpenSSH-Server, con un usuario/clave ó clave exportada al equipo PCSniifer.

Dispondrá además de un script para el rotado de ficheros. Para ello se ejecutará cada hora, en minutos “valle” (los de menos uso de CPU), de modo que se elimine el fichero más antiguo existente.

A continuación, y debido a su simplicidad, se incluyen los códigos a aplicar en crontab y el script de rotado.

```
#sudo nano crontab

30 *  * * * /home/snipol/snipol/scripts/rotateCaptures /home/snipol/captures
```

```
#!/bin/bash
ls -t1 $1 | tail -1  | xargs  rm
```

Capítulo 6

Implementación e implantación del software

6.1 PROCESO DE CODIFICACIÓN

6.1.1 Clases.

De manera somera se presentan las siguientes clases y una descripción de sus tareas:

CLASE	DESCRIPCIÓN
BloomFilter	Lógica de funcionamiento del algoritmo de Bloom filters.
DataBaseSqlite3	Rutinas para la creación de alarmas en una base de datos Sqlite 3.
hl_exception	Excepciones existentes al realizar un hash SHA1
hl_sha1	Código para la generación de hashes SHA1
hl_sha1wrapper	Interfaz c++ para el uso de SHA1
hl_types	Clases de SHA1 existentes según su longitud.
Indect3	Inicio del plugin de alertas.
processNewLastdataFile	Procesamiento de un fichero índice de ficheros con información decodificada.
processFile	Lógica para procesar un sólo fichero decodificado.

6.1.2 Algoritmos

- **Librería para Bloom Filters**: se utilizará una implementación existente del algoritmo de Bloom filters. Actualmente las versiones más conocidas son:

  - Arash Partow
    http://code.google.com/p/bloom/
  - Dean Michael Berris
    https://svn.boost.org/svn/boost/sandbox/bloom_filter/trunk/boost/bloom_filter/

Los datos resumidos y estadísticos sobre ambas implementaciones son:

Característica	Arash Partow	Dean Michael Berris
Versión actual	1.21	1.00
Fecha creación	03/26/09	06/08/09
Resultados en Google	1880 [1]	1120 [2]

Al ser la más difundida y revisada, se utiliza la versión de Arash Partow.

1 http://www.google.es/search?hl=en&source=hp&q="Partow"+bloom+filters&btnG=Google+Search&aq=f&oq=

2 http://www.google.es/search?hl=en&source=hp&q="Berris"+bloom+filters&btnG=Google+Search&aq=f&oq=

- **Librería para SHA1 hashing**: se utilizará la implementación abierta de *http://hashlib2plus.sourceforge.net/*

  En posteriores versiones, se reemplazará por la tecnología Fuzzy Hashing, implementación "**DeepToad**"

6.1.3 Threads

6.1.3.1 xplicoAlerts

Para mejorar el rendimiento se implementará el uso de threads para procesar los ficheros de índices “lastdata.txt” generados por Xplico.

6.1.3.2 Envío

Se implementará un thread dedicado al efecto. Tomará una captura de una hora ya pasada y no procesada, la comprimirá con “pigz” y la enviará. Se ejecutará cada hora.

6.1.3.3 Semáforos

Se estudiará el uso de semáforos, mayoritariamente en los casos de modificación dinámica de los conjuntos de información que educan los Bloom filters.

6.1.3.4 Compilación

Se compilarán con un script “compilar.sh” ó el siguiente comando:

```
# g++ -o"xplicoAlerts" -lsqlite3 -lboost_date_time -lboost_filesystem DataBaseSqlite3.cpp hl_sha1.cpp hl_sha1wrapper.cpp lndect3.cpp processFile.cpp processNewLastdataFile.cpp -Wall
```

6.1.3.5 MPI

Se considerará, si fuese necesario, el uso de tecnologías MPI (como *OpenMPI ó MPI.NET)* para posibilidad el multiproceso, de cara a mejorar el rendimiento notablemente.

En este apartado se describe la metodología a seguir para la instalación del software. Este proceso se irá refinando, mejorando y simplificando en posteriores revisiones del documento, con la finalidad de hacerlo más amigable al usuario.

La instalación se compone de los siguientes pasos, ejecutados todos con el usuario “root” :

6.2.1 Instalación del sistema operativo en PC_Sniffer

1.- Descargar e instalar Debian 5.0.3
http://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/

Nota: El sistema se ha testeado en Debian 5.0.3, pero otras versiones de GNU/Linux pueden ser compatibles.

La máquinas debe tener dos interfaces de red y deben ser configuradas en dos redes distintas.

2.- Una vez instalado, reiniciar la máquina.

3.- Desinstalar los siguiente paquetes:

A incluir en posteriores revisiones del documento

4.- Descomentar los repositorios que vengan comentados en /etc/apt/sources.list

5.- Actualizar el sistema:

```
# su -
# apt-get update
# apt-get dist-upgrade
```

Nota: esta operación instalará versiones actualizadas del software preexistente, teóricamente compatible con este software, y sin apenas riesgo de producir problemas. Debido a la evolución constante del software, si lo desea puede consultar mediante el foro del proyecto Xplico.org la conveniencia de realizar una actualización.

6.- Configurar el sistema para que no haga ninguna conexión automática con el exterior, como la descarga de nuevas actualizaciones ó ntp.

A incluir en posteriores revisiones del documento

7.- Reiniciar la máquina.

```
# reboot
```

8.- Realizar las operaciones de blindaje del sistema:

A incluir en posteriores revisiones del documento

6.2.2 Instalación de paquetes relacionados

1.- Instalar los siguientes paquetes:

```
# su -
# apt-get update
# apt-get install sqlite tcpdump tshark apache2 php5 php5-sqlite build-essential perl zlib1g-dev libpcap-dev libsqlite0-dev libmysqlclient16-dev php5-cli libapache2-mod-php5  libx11-dev libxt-dev libxaw7-dev python-all sqlite3

# apt-get install libsqlite3-dev libboost-filesystem1.40-dev libboost-date-time1.40-dev libboost-date-time1.40.0 libboost-filesystem-dev gcc-4.4
```

2.- Testar el funcionamiento de tcpdump, Apache, Sqlite3:

```
# root@virtuakarmic:/opt/xplico# /etc/init.d/apache2 status
 * Apache is running (pid 9027).

# root@virtuakarmic:/opt/xplico# tcpdump -s0 -v
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
CTRL+C

# sqlite3
.exit;
```

3.- Crear clave ssh.

A incluir en posteriores revisiones del documento

```
# ssh-keygen -t dsa -b 1024 -f id_dsa_something -C 'Some comment'

This creates a DSA key with 1024 bits of random key material, stores the secret key in a file called "id_dsa_something", and stores the public key in a file called "id_dsa_something.pub" with the indicated comment. The key length, 1024 bits, is currently the "standard" length. I have used keys with 2048 bits, and the ssh-keygen program will accept anything as high as 32768 bits. 4096 bits is considered "military grade", anything above that is probably overkill, since the key isn't protecting the session traffic, only the negotiations for the session key.
```

Y realizar su exportación entre ambos equipos.

4.- Instalación de Xplico

Descargar el instalador de Xplico 0.54 en formato ".deb" e instalarlo

```
# wget http://sourceforge.net/projects/xplico/files/Xplico%20versions/version%200.5.4/xplico_0.5.4_i386.deb/download

# gdebi  xplico_0.5.4_i386.deb
```

Nota: si se desea el binario de la arquitectura 64 bits, es necesario seguir las instrucciones oficiales de Xplico.org: http://wiki.xplico.org/doku.php?id=tutorial:0.5.5

Copiar el fichero de configuración para Apache2:

```
# cp /opt/xplico/cfg/apache_xi /etc/apache2/sites-enabled/xplico
```

Añadir en el fichero de Apache el puerto de Xplico:

```
# nano /etc/apache2/ports.conf
(...)
#xplico Host port
NameVirtualHost *:9876
Listen 9876
(...)
```

Modificar las siguientes variables de PHP a los siguientes valores recomendados:

```
# nano /etc/php5/apache2/php.ini.
(...)
post_max_size = 100M
upload_max_filesize = 100M
(...)
```

Activar el módulo "rewrite" en Apache:

```
# a2enmod rewrite
```

Reiniciar Apache2:

```
# /etc/init.d/apache2 restart
```

Ejecutar Xplico:

```
# /etc/init.d/xplico start
```

4.- Testeo del funcionamiento de Xplico.

Descargar y descomprimir la siguiente captura de tráfico:

```
http://wiki.xplico.org/lib/exe/fetch.php?media=pcap:xplico.org_sample_capture_protocols_supported_in_0.5.4.pcap.bz2
```

Crear un caso en Xplico y procesar la captura. Se deben obtener varias informaciones en cada protocolo decodificadas.

6.2.3 Instalación del software xplicoAlerts.

En sucesivas revisiones de este software se incluirá un script instalador

1.- Descomprimir el paquete

```
# tar xvfz xplicoAlerts-1.0.0.tar.gz
```

2.- Copiar el binario al directorio de binarios de Xplico.

```
# cp xplicoAlerts /opt/xplico/bin/
```

3.- Crear las nuevas tablas en la base de datos.

```
# sqlite3 /opt/xplico/bin < create.xplicoAlerts.tables.sql
```

4.- Añadir la interfaz gráfica.

```
# cp -R xplicoAlertsGUI/* /opt/xplico/xi/
```

5.- Añadir permisos de “sudo”.

```
# echo “www-data ALL=(root) NOPASSWD: /usr/bin/killall” >> /etc/sudoers
# echo “www-data ALL=(root) NOPASSWD: /opt/xplico/bin/Indect3” >> /etc/sudoers
```

6.- Creación de la tubería/pipe de comunicación del servidor web con el binario.

6.2.4 Verificación rápida de funcionamiento

A incluir en posteriores revisiones del documento.

6.2.5 Instalación del PC_Operador

1.- Repetir paso “Instalación del sistema operativo en PC_Sniffer”

2.- Instalación de interfaz gráfico y herramientas:

```
# apt-get install gnome openssh-server firefox wireshark
```

3.- Reiniciar:

```
# reboot
```

Capítulo 7

Software generado

El software generado en esta etapa realiza la función de comprobar contra una base de datos, mediante uso de Bloom filters, la pertenencia o no de los ficheros transmitidos a una lista negra de ficheros sospechosos.

Instalado el software, en esta etapa se pueden realizar las siguientes funcionalidades:

1º) Altas/bajas/modificaciones de categorías o crímenes: uso de crímenes para poder especializar la búsqueda de contenidos sospechosos en un usuario montorizado.

2º) Altas/bajas/modificaciones de hashes de ficheros: inclusión de hashes SHA1 para poder utilizarlos como sospechosos en la búsqueda de contenidos maliciosos.

3º) Bajas/modificaciones parciales de alertas: visualización de las alertas generadas, con la posibilidad de incluir notas por parte del operador para facilitar su trabajo.

4º) Arranque de la aplicación: arranque y parada de la aplicación mediante el interfaz web, de cara a facilitar su interacción con el software y hacerlo lo más extensible a cualquier perfil de usuario policial.

Quedando para posteriores fases el desarrollo de otras funciones, principalmente:

- Configuración.
- Detección de palabras sospechosas.
- Detección de conexiones a hosts sospechosos.

Se mantiene una demo de la aplicación en la siguiente URL:

http://163.117.140.5:9876

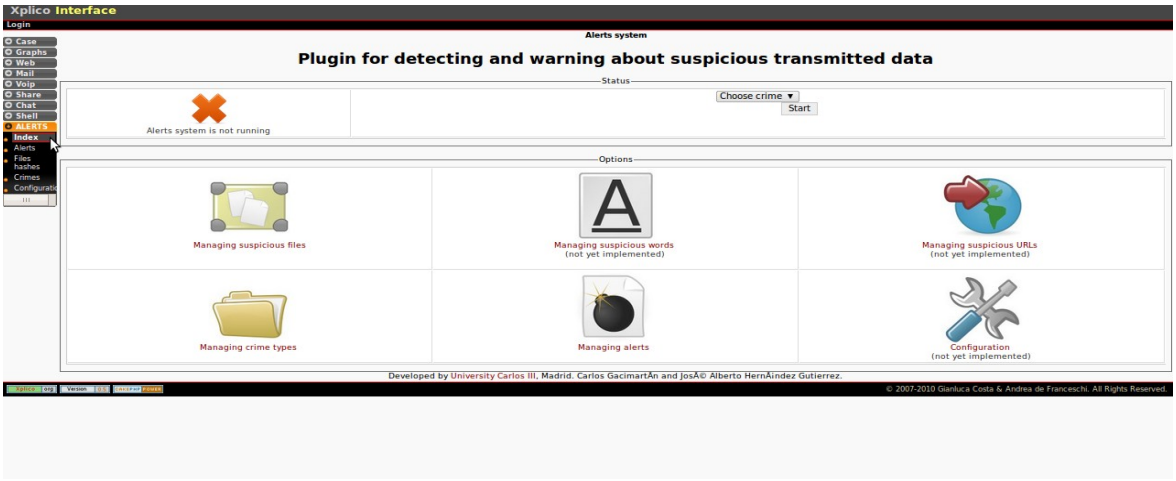
usuario y contraseña: “xplico”

El interfaz de Xplico permite crear dos tipos de caso:

- Live acquisition: captura en tiempo real del tráfico del interfaz seleccionada de la máquina donde se ejecuta Xplico.

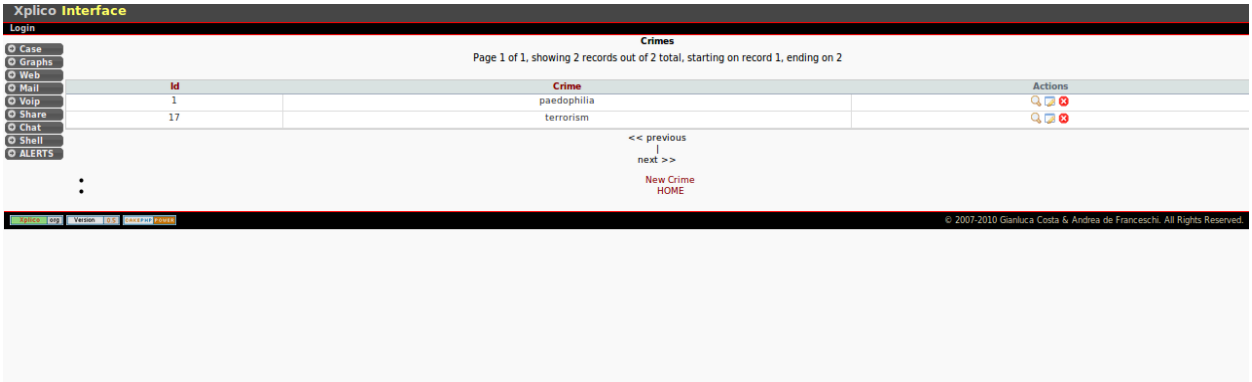
– Upload PCAP file: procesar los datos tomados previamente en una captura de tráfico en formato PCAP 2.4

Una vez seleccionada la opción deseada, es necesario activar el plugin xplicoAlerts. Para ello, seleccionar las opciones “Alerts > Index”.

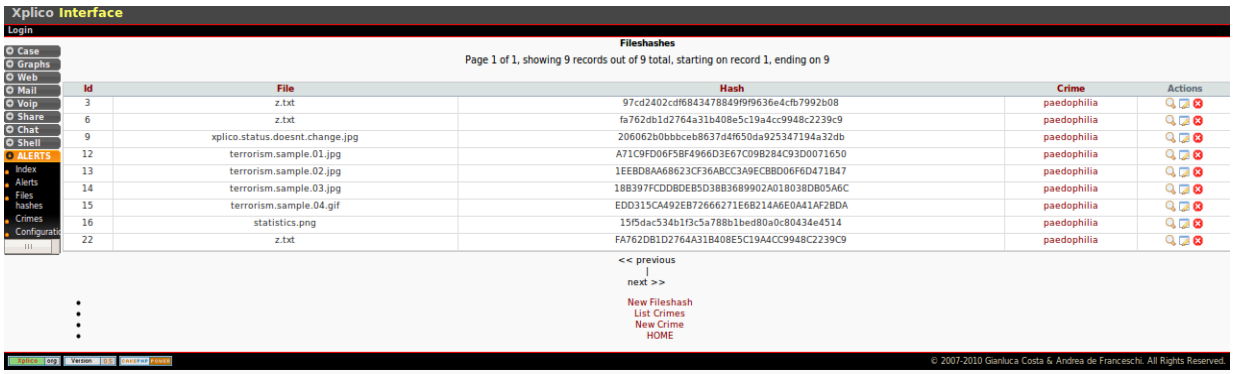


Elegir un tipo de crímen y hacer click en “Start”. El sistema de alertas estará activo, todo tráfico procesado tanto en tiempo real o por ficheros de capturas subidas al software de Xplico será analizado en busca de información sospechosa.

En la opción “Managing crimes” se pueden crear facilmente los crímenes a perseguir. Opcionalmente se puede subir en ese mismo momento un fichero TXT con un hash por línea que se asociará a este crímen.

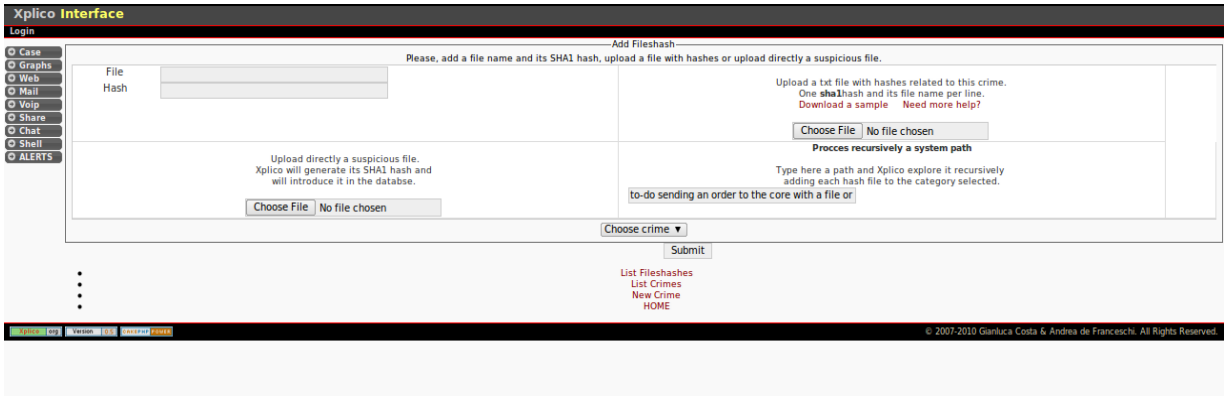


En la opción “Managing suspicious files" se pueden ver los hashes existentes, así como borrarlos o cargar nuevos en el sistema.

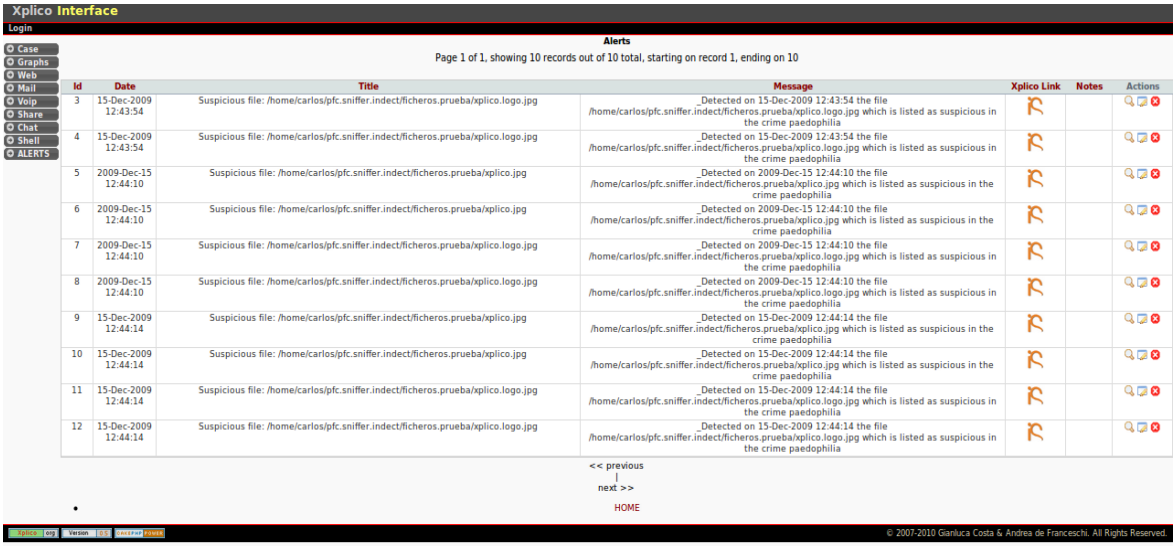


Para facilitar el uso lo máximo posible al usuario, se han dispuesto 4 maneras de incluirlos:

- Añadir un sólo hash y su nombre de fichero (opcional).
- Subir un fichero txt con un hash por línea.
- Subir un fichero sospechoso, del cual el sistema calculará su hash, lo incluirá en la base de datos y eliminará el fichero.
- Procesar una ruta del sistema: indicar un path del sistema donde se sitúan ficheros y directorios, recorrerlo recursivamente calculando el hash de cada fichero e ir añadiéndolo al sistema.



Visualización de alertas: permite ver un listado de las alertas, ordenadas cronológicamente (la más reciente la primera, facilitando el uso al operador), editarlas (añadiendo información, nunca eliminándola) y eliminarlas.



Ejemplo de uso:

1.- Descargar la siguiente captura de tráfico:

http://wiki.xplico.org/lib/exe/fetch.php?media=pcap:xplico.org_sample_capture_images.pcap

2.- Incluir el siguiente nuevos hash como sospechoso de un crímen:

9a7719d855f1959dc11e50e60c2c8e090fc291cd

2.- Arrancar el motor de xplicoAlerts seleccionando el mismo crímen que en el punto 2.

3.- Crear un caso “PCAP upload” en Xplico.

4.- Subir la captura de datos.

5.- Visualizar en “Alerts” la alarma generada.

Capítulo 8

Implicaciones legales del proyecto

Debido a las implicaciones legales y tipo de información obtenida con esta utilidad, el uso de este software está restringido a las Fuerzas y Cuerpos de Seguridad del Estado bajo orden judicial.

Así mismo se deberán atender a las indicaciones de las leyes de protección de datos de cada país aplicadas a las intervenciones por las Fuerzas de Seguridad.

Será conveniente requerir asesoramiento de la Agencia de Protección de Datos. Así mismo se cuenta con el asesoramiento legal de la Unión Europea, impulsora principal de este proyecto.

Capítulo 9

Conclusiones y líneas futuras

El alcance de este proyecto es amplio, y su duración está prevista que se prolongue **hasta 2012**. Esto significa que, utilizando como base este documento de líneas generales y prototipo, se posibilitará realizar una tecnología adecuada a las necesidades de las fuerzas policiales que irán testando y facilitando feedback al proyecto.

Los objetivos propuesto en este proyecto se han superado satisfactoriamente; se ha realizado un planteamiento global de una herramienta a desarrollar y se ha realizado la implementación del primer módulo, el correspondiente a la detección de ficheros transmitidos.

En resumen, actualmente esta aplicación proporciona detección de ficheros sospechosos con una aplicación práctica en la detección fundamentalmente de la pedofilia, pero la potencia de la infraestructura presente facilita la posibilidad de nuevos métodos de detección de amenazas. Por ello a lo largo de los próximos meses se realizarán desarrollos y pruebas intensivas que garanticen el máximo funcionamiento de esta aplicación y cumplimiento de sus objetivos.

9.1 Conclusiones sobre el proyecto

9.1.1 Dificultad del Proyecto

Hasta la situación actual se han superado los siguientes escoyos:

- Uso de un software que extraiga la información encapsulada en protocolos. Para ello se valoraron varias alternativas, inclinándose la decisión final por Xplico.
- Funcionamiento en tiempo real: dada la naturaleza de la actividad, la obtención de información sobre amenazas es vital, y se ha de conseguir en el mínimo tiempo posible para poder reaccionar a tiempo. Para ello se han utilizado las tecnologías más avanzadas y se han realizado pruebas de laboratorio, a espensas de las pruebas reales, obteniéndose alarmas en tiempo real.
- Diseñar el sistema de modo que no se pueda dar cuenta el espiado: por lo que se ha optado por la arquitectura en modo sniffer frente a bridge, y tecnologías que en ningún modo puedan alterar el funcionamiento de terceros. Así mismo se ha optado por blindar el prototipo de modo que sea prácticamente invulnerable, con los medios actuales, a su disección.

9.1.2 Resultado obtenido

Producto	Responsable	Versión
Xplico	Gianluca Costa, Andrea De Franceschi Carlos Gacimartín - UC3M	0.5.5
XplicoAlerts	Carlos Gacimartín - UC3M	0.0.1

xplicoAlerts, al estar basado en software GPL como Xplico, Cakephp y otros, y debido a la naturaleza “vírica” de esta licencia, está obligado a licenciarse como **GPL**. El equipo de Xplico ha aprobado su inclusión en la rama principal de desarrollo. Esto facilitará y generará un gran crecimiento de ambas herramientas, mientras que no facilitará a las organizaciones criminales descubrir maneras de eludir su supervisión al no hacerse públicos los conjuntos de entrenamiento del sistema. Así mismo, la publicación del proyecto en la biblioteca de la universidad lo hacen accesible públicamente.

Por otra parte, el proyecto presentado en este documento es resultado de la primera fase del proyecto. Posteriormente, y como fruto de tesis, se presentará aproximadamente en 2011 el resto de fases.

El desarrollo de este sistema ha implicado la directa generación de las siguientes contribuciones al software libre:

- Xplico.org: Forum (http://forum.xplico.org). Creación de un foro para obtener todo el feedback posible de los usuarios.
- Verificación de la documentación de Xplico (http://wiki.xplico.org): Revisión de los manuales de instalación, con sugerencias, mejoras y correcciones.
- Generación de paquete .deb de Xplico.
- Generación de nueva documentación en el wiki de Xplico.org
- Creación de un bugtracker y “features request” para el proyecto Xplico: http://sourceforge.net/tracker/?group_id=239471&atid=1110205
- Programación de un servicio o daemon para Xplico.
- Varios bugs en el funcionamiento de Xplico detectados, reportados y posteriormente solucionados.
- Generación de 23 how-to's públicos sobre diversos issues.

9.1.3 Otras conclusiones

- Aplicación de experiencia y conocimiento en aplicaciones cerradas en este proyecto.
- Aplicación de este proyecto en la docencia.
  - Visualización de la información extraida de una captura.
  - Demostración de la necesidad de utilizar tráfico cifrado, al obtenerse con este software la mayoría de las contraseñas de manera sencilla.
  - Posibilidad de ofertar otros proyectos como ampliación de este proyecto para el alumnado: creación de “dissectors” de protocolos que aun no estén soportados en Xplico.
- Continuación en este proyecto incluso en el tiempo libre.

9.2 LÍNEAS FUTURAS

En un posterior desarrollo de este software sería interesante y probablemente bastante beneficioso, focalizar en las siguientes actividades:

A corto plazo, se plantea el desarrollo de los módulos de análisis de contenido, de modo que se busquen cadenas de textos sospechosas dentro de todos los ficheros. Así mismo se propondrá una inteligencia artificial para evaluar la importancia de esas palabras de cara a generar o no alarmas.

También se estudia un segundo módulo para analizar las URLs o hosts con los que se comunica un sospechoso, y viceversa, el análisis de conexiones que tiene un servidor.

Así mismo se espera presentar el primer prototipo ante las autoridades norirlandesas en Marzo, el cual estará más maduro que el presente, y significará una gran inyección de experiencia en el proyecto, así como posibilitará mantener una línea directa con la organización para poder fomentar el uso de tests y ganar más conocimiento de la temática para proyectarlo en la aplicación.

Uno de los pilares fundamentales para hacer viable esta solución es apoyar al proyecto de software libre Xplico.org, el canal a través del que se obtiene la información capturada en un formato manejable. El crecimiento de este proyecto implicará nuevos protocolos soportados y por ende más información analizable, que en algunos casos derivará en alertas que antes pasaban desapercibidas, y en otros aportará informción complementaria a alertas existentes, lo cual aumentará la productividad de este sistema.

Por último, una de las pautas fundamentales en la actual metodología de búsqueda de amenazas es el cruce de datos. Para ello se implementarán sistemas que faciliten la sencilla exportación de datos para formar una visión completa del sospechos y entorno monitoreados.

Testeo en situaciones simuladas así como reales, de cara a obtener feedback tanto del Operador en su modo de uso como en los resultados que pretende obtener, como en su adecuación e instalación en diversos escenarios.

Mejora de la plataforma:

- Hardware: a este nivel, reevaluar todos los componentes analizados, haciendo epecial himcapié en la mejora de CPU, disco duro e interfaz de enlace entre PCSniffer y Operador, dotándolo de un mayor ancho de banda.
- Software: actualización de las actuales versiones de software dentro de la misma serie, en busca de mejora de rendimiento y solución de posibles problemas.
- Inclusión de un segundo canal de comunicación entre el PCSniffer y el Operador para paliar los problemas de velocidad, a la vez que se ofrece una garantía mayor de funcionamiento.
- Modificación del Interceptor para tomar, desde una segunda salida, el tráfico analógico de la línea. Procesarlo a través de un modem analógico instalado en el PC_Sniffer y realizar grabaciones telefónicas en mp3 u ogg, ofreciendo un sistema completo para la **interceptación de las comunicaciones de una línea telefónica** analógica.
- Creación de interceptores para otros tipos de conexiones de banda ancha: fibra óptica, 3g.
- Clustering: en situaciones excepcionales en las que se manejen grandes cantidades de datos, mayores de las habituales, habilitar un sistema de computación distribuida para cubrir las necesidades de cómputo.
- Uso de tecnología “PF_RING/nPRobe” para adquisición de datos en redes.
- Homologar el aparato por una entidad externa (ISO).
- Soportar autoaprendizaje en el análisis léxico de ficheros: variar el valor de los pesos de palabras que estén cerca de otras catalogadas como sospechosas.
- Inclusión de soporte para esteganografía básica.
- Incluir funciones de descompresión de ficheros y obtención de passwords de los mismos si procediese.
- Conexión con bases de datos públicas de spammers, proxys y otros para completar la información dispuesta.

Capítulo 10

Bibliografía y referencias

10.1 BIBLIOGRAFÍA Y RECURSOS ELECTRÓNICOS DE REFERENCIA

Para la redacción del presente documento se ha consultado documentación sobre la metodología MÉTRICA Versión 3.

Estos documentos de planificación, desarrollo y mantenimiento de sistemas de información, propiedad del Ministerio de Administraciones Públicas, pueden consultarse en la página Web que aparece a continuación:

http://www.csi.map.es/csi/metrica3/

Sobre bibliografía, se han consultado las siguientes publicaciones:

- *"Introducción a la ingeniería de software"*, José A. Cerrada,Manuel Collado.
- *"Network Applications of Bloom Filters: A Survey",* Andrei Broder y Michael Mitzenmacher

Así mismo se han utilizado los siguientes recursos online:

- http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/ch-ssh.html
- http://arstechnica.com/open-source/news/2009/01/super-fast-ext4-filesystem-arrives-in-ubuntu-9-04.ars
- *http://d-i.alioth.debian.org/manual/es.i386/ch02s01.html*
- http://es.wikipedia.org/wiki/LUKS
- http://www.saout.de/tikiwiki/tiki-index.php?page=LUKS
- *http://oei.yungchin.nl/2008/04/23/installing-ubuntu-804-with-full-disk-encryption/*
- *RFC 3174, Algoritmo para claves hash SHA1* http://www.faqs.org/rfcs/rfc3174.html
- *http://www.seraphinux.com/index.php/102/2008/04/30/sqlite-una-base-de-datos-embebida/*
- *Referencia comparativa c++ / Mono c#]*
  http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=gpp&lang2=gpp
  http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=csharp&lang2=csharp

10.2 DEFINICIONES

DEFINICIONES BÁSICAS	
dummies	En programación de software, pequeños módulos o programas que generan datos que un elemento está esperando recibir de una entidad externa, de modo que se puede simular y probar su funcionamiento.
Sniffer	programa de captura de las tramas de red.
DSL	"Línea de Suscripción Digital Asimétrica".
Paquete	Unidad de datos a nivel IP en el modelo OSI.
Trama	Unidad de datos a nivel de red en el modelo OSI.
Payload	Información de usuario.
Decodificar	Extraer de capturas de tráfico la información que se pretende transmitir.
Analizar	Buscar en información *decodificada* palabras susceptibles de sospecha.
How to	Término inglés para definir cómo realizar una tarea de manera sencilla.

10.3 ACRÓNIMOS

ACRÓNIMOS	
WP3	Work Package 3 - INDECT
WP4	Work Package 4 - INDECT
SW	software
HW	hardware
SO	Sistema operativo.
HPD	Visitas diarias (Hits Per Day)
ISP	Proveedor de servicios de Internet
SNR	Signal Noise Ratio.
BD	Base de datos
BF	Bloom filters

Anexo I: "Gestión del proyecto"

1. Medios técnicos empleados para el proyecto

Para el desarrollo de este proyecto se ha utilizado el siguiente material:

HARDWARE	
CPU	2.4 GHZ
Memoria RAM	4GB
HD	500 GB
Monitor	22"
Tarj. red	Nvidia Corporation MCP55 Ethernet (rev a2) 10/100/1000

SOFTWARE	
SO	Ubuntu 8.04.2
Software ofimático	OpenOffice 3.1
Virtualización	VirtualBox 3.06
Planificación de proyectos	Planner 0.14.4
SO virtualizado	Debian 5.0
IDE C++	Eclipse Galileo C++ plugin
Navegador web	Firefox3.5.1

MÁQUINA PROTOTIPO LAB. UC3M	
CPU	Core 2 Duo 2.66 GHz
Memoria RAM	3GB
HD	2TB
SO	Ubuntu 9.04 - Jaunty
Red	10/100/1000

2. Análisis económico del proyecto

Se expone una **preestimación** de los costes del proyecto. Según la estimación de horas realizada por el equipo, y a un coste de 30€/h de ingeniero, se estiman los siguientes cálculos en este preanálisis:

Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost
1	Planificación	Sep 1	Sep 2	2d		0%	480
2	**Obtención requisitos**	Sep 3	Sep 4	2d			480
2.1	Req. HW	Sep 3	Sep 3	1d		0%	240
2.2	Req. SW	Sep 4	Sep 4	1d		0%	240
3	Análisis	Sep 7	Sep 11	5d		0%	1,200
4	Planificación global	Sep 14	Sep 14	1d		0%	240
5	Diseño arquitectónico	Sep 15	Sep 21	4d 3h		0%	1,057.5
6	Diseño detallado	Sep 21	Sep 29	6d 4h		0%	1,582.5
7	Prototipo no funcional	Sep 30	Oct 2	3d		0%	720
8	Diseño pruebas	Oct 5	Oct 6	1d 6h		0%	435
9	Revisión fases anteriores	Oct 6	Oct 9	3d 1h		0%	765
10	Implementación	Oct 12	Nov 20	30d		0%	7,200
11	Integración	Nov 23	Dec 18	20d		0%	4,800
12	**Pruebas**	Dec 21	Jan 15	20d			4,800
12.1	Integración	Dec 21	Jan 1	10d		0%	2,400
12.2	Validación	Jan 4	Jan 8	5d		0%	1,200
12.3	Aceptación	Jan 11	Jan 15	5d		0%	1,200
13	Reunión fin del proyecto	Jan 18	Jan 20	2d 2h		0%	547.5

TOTAL: 24,307.5

2.1.1 Presupuesto inicial

Ningún precio o cifra económica de este documento incluye IVA o impuesto aplicable.

COSTE MATERIAL		
TAREA	**CANTIDAD**	**COSTE**
PC AMD2000 – 4GB -	810h	210 €
PC servidor (svn, backup)	810h	340 €
PC Laboratorio Telemática	810h	210 €
Cds para backups.	20	10,00 €
Dietas (reuniones y desplazamientos)	5	100,00 €
Licencias SO Debian	3	0.00 €
Licencias Xplico.org	3	0.00 €
Licencias tcpdump	3	0.00 €
Licencias xplicoAlerts	3	0.00 €
Licencias scp	3	0.00 €
TOTAL		**870 €**

Nota: hardware utilizado imputado proporcionalmente a su uso.

Reseñar la ausencia de coste de licencias de software al desarrollarse todo bajo software libre.

TOTAL COSTE DESARROLLO HARDWARE	
CONCEPTO	**COSTE**
Interceptador	N/A
Router	N/A
PC-análisis	1000 €
PC-Servidor	500 €
TOTAL	**1.500 €**

TOTAL COSTE DESARROLLO SOFTWARE	
CONCEPTO	**COSTE**
Coste humano	24307.5
Coste material	870 €
Coste hardware	1.500 €
TOTAL	**26.677,5**

2.1.2 Presupuesto final

Fuentes de riesgo:

La experiencia dice que el 80% del riesgo total de un proyecto se debe solamente al 20% de los riesgos identificados. Después de un trabajo iterativo entre esta fase y el análisis, se determinan como principales riesgos:

- **Variación de los requisitos por parte del cliente**: al no especificarse requisitos concretos por parte de la documentación de Indect, puede interpretarse tanto positiva como negativamente. Por una parte, hay libertad para adaptar el producto a la visión del desarrollador (uc3m) pero por otra, una malinterpretación de objetivo de INDECT en conjunto puede llevar al fracaso total este proyecto.

- **Retrasos por parte del usuario final para realizar las pruebas**: la no disponibilidad de una autoridad que certifique o valide este sistema como adecuado a los requisitos puede implicar una demora en el tiempo nada conveniente.

- **Hardware no compatible con la tecnología planteada**: en el mercado existe una gran cantidad de tecnología disponible, tanto hardware como software. Una descoordinación entre ambos elementos supondría el no funcionamiento del sistema.

Por ello se consideran en base a la experiencia un 20% de Riesgo.

Presupuesto software para cliente = coste + riesgo .

CONCEPTO	%	Cantidad
Coste	100,00%	26.677,5 €
Riesgo	20,00%	5335.5 €
TOTAL		**32.013 €**

Anexo II: “Tareas periféricas cuya prioridad es necesaria calcular”.

- Definir idiomas suplementarios y traducir la aplicación a esos idiomas.
- Realizar paquete instalador .deb (Debian) y .rmp (Fedora) de Xplico con xplicoAlerts para promocionar su uso, generando valioso y gratuito feedback entre usuarios.
- Enviar los paquetes instaladores a los repositorios oficiales de las distintas distribuciones.
- Generación de papers sobre la utilidad, experiencias y conocimiento generado en este proyecto.
- Establecimiento de líneas de trabajo estrechas con los usuarios de Indect para facilitarles periodicamente un prototipo (máquina virtual) que puedan evaluar.
- Creación de CD con una distribución Debian modificada con el sistema preinstalado, lista para instalar en equipos. Ej: http://softlibre.barrapunto.com/softlibre/09/08/03/213202.shtml
- Investigar, en la medida de lo posible, otros proyectos como Echelon, Sitel o Carnivore para obtener ideas aplicables a este proyecto.
-
- Investigar y desarrollar una mejora para Xplico de cara a facilitar el rotado de información y/o evitar el colapso de los medios de almacenamiento.
- Verificación de funcionamiento y benchmarks en redes Ipv6.
- Descomprimir archivos comprimidos y descubrir clave si procede.
- Testear el sistema en redes Imagenio.
- Realizar tests de rendimiento en otras plataformas: Fedora, CentOS, SuSE y similares.
- Aplicar un segundo software de decodificación, como Pyflag ó Nast, para los flujos no reconocidos por Xplico.
- Inclusión de los algoritmos Fuzzy Hashing.
- Estudiar la conveniencia de aplicar cifrado en la base de datos.
- Inclusión de un autorefresh en la sección de alertas para facilitar la comunicación de las mismas.
- Creación de nuevos sospechosos, para no mezclar alertas.
- Firmado de capturas.
- Cambio de prioridad de los procesos Xplico y xplicoAlerts por interfaz web.
- Generación de un paper sobre comparativa de sniffers: rendimiento, utilidades y limitaciones. (paper de pago: http://portal.acm.org/ft_gateway.cfm?id=1047873&type=pdf).
- Generación de tesauros temáticos sobre distintos crímenes para entrenar los filtros, solicitando colaboración a la asignatura “Ingeniería de la información” de Ing. Informática.
- Homologar el sistema por una entidad externa.
- Desarrollar un prototipo con tecnología Coreboot, tanto en entornos virtuales como reales, evaluando su rendimiento.

- Realización de las sugerencias de mejora del proyecto Xplico propuestas en:

  https://sourceforge.net/tracker/?group_id=239471&atid=1110205

2934963	Xplico web interface using HTTPS	2010-01-19
2933805	Removing test cases of Cakephp to save space	2010-01-17
2918884	Doing a solution to have an enhanced realtime capture method	2009-12-21
2918842	Stats of performance	2009-12-21
2917640	Set of example captures	2009-12-19
2917639	demo online	2009-12-19
2917637	Develop packages for binary and XI	2009-12-19
2917096	Starting dema/xplico binary from web interface (XI)	2009-12-18
2917093	Monit - Create a config example	2009-12-18
2916955	Doing a manual for Xplico - TNAPI	2009-12-18
2915213	Creating a script for installing Xplico easily	2009-12-16
2914130	support for Pcap-ng	2009-12-14
2913356	[legal] add a reference of a software used	2009-12-12
2913351	adding this icon for pcap downloads in tables	2009-12-12
2913287	Decoding skype webchat	2009-12-12
2909368	Checking updates	2009-12-05
2909357	Create xplico deb own repository	2009-12-05
2909356	Submit .deb package to Debian and Ubuntu repositories	2009-12-05
2909059	Decoding most popular webmails	2009-12-04
2909057	Decoding Jabber/Gtalk	2009-12-04
2909056	Decoding Ms. Messenger (MSN)	2009-12-04
2909055	Decoding (recording) VOiP calls	2009-12-04
2909053	Decoding SSL traffic (with keys)	2009-12-04
2909026	VNC and Terminal Server	2009-12-04
2908956	SEO work	2009-12-04
2908910	NTP dissector	2009-12-04
2908905	p2p emule dissector	2009-12-04
2908902	SSH dissector	2009-12-04
2908901	Mysql or PostgreSQL compatibility	2009-12-04
2908895	Update wiki documentation - Apache resources	2009-12-04
2908893	Upgrade XI to cakephp 1.2.6	2009-12-04
2908891	Add function getMyPid() to Alerts system	2009-12-04
2908886	Updating the third-party files	2009-12-04
2908884	RPM	2009-12-04
2908883	relax version of requested packages	2009-12-04
2908880	Integrate alerts system	2009-12-04
2908877	autovacuum at the database	2009-12-04
2908873	Allow modifying Xplico xplico.cfg using the XI	2009-12-04
2908871	Option for rotating contents	2009-12-04
2887452	man xplico	2009-10-27
2881759	Suggestion: change priority of Xplico	2009-10-19
2881693	rt_demo.sh Xplico binary: -m parameter has changed	2009-10-19
2881690	tcpdump permission denied	2009-10-19
2881655	Managing users	2009-10-19
2881653	Managing cases and sessions	2009-10-19
2881635	Support for multilanguages	2009-10-19

Anexo III: "Catálogo de requisitos".

1.- Formato para la especificación de requisitos

En el presente apartado se identificarán los requisitos para el sistema.En primer lugar se expondrá el formato escogido para la presentación de cada uno de estos requisitos, que se recogerán en tablas como la que se incluye a continuación:

IDENTIFICADOR: R-[T] [N]	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☐ BAJA	**FUENTE:**
NECESIDAD: ☐ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	
DESCRIPCIÓN:	

Detallándose cada uno de los campos que conforman la tabla anterior de la siguiente manera:

- **Identificación**: cada requisito de usuario incluirá una identificación, para facilitar su seguimiento y el rápido acceso a éste. El identificador de cada requisito se construirá según la sintaxis SR- [T] [N], donde el campo [T] se corresponde con las siglas que identifican cada tipo concreto de requisito software y el campo [N] está compuesto de tres dígitos que recogen el orden que ocupa el requisito de entre los especificados para su mismo tipo.

  Los tipos concretos de requisitos que se contemplará, enmarcados dentro de los requisitos software, y las siglas que se asociarán a los mismos serán:

SIGLAS	TIPO CONCRETO DE REQUISITOS
OP	Requisitos de operación
I	Requisitos de información
RC	Requisitos de restricción a las capacidades del sistema
REN	Requisitos de rendimiento
INT	Requisitos de interfaz
REC	Requisitos de recursos
VER	Requisitos de verificación
PA	Requisitos de pruebas de aceptación
DOC	Requisitos de documentación
SSA	Requisitos de seguridad del sistema frente a amenazas
POR	Requisitos de portabilidad
PEU	Requisitos de prevención de errores del usuario
ME	Requisitos de manejo de errores
MAN	Requisitos de mantenimiento
SPF	Requisitos de seguridad de las personas frente a fallos
INV	Requisito inverso

- **Necesidad**: los requisitos esenciales para el sistema; el resto pueden ser menos importantes. Resulta importante que mediante aquellos requisitos que se indiquen como esenciales se puedan cubrir todas las funcionalidades requeridas por el usuario, especificadas a través de los requisitos de usuario.

- **Prioridad**: cada requisito de software incluirá una medida de la prioridad que posee, para que el desarrollador pueda decidir la planificación del desarrollo.

- **Estabilidad**: algunos **requisitos** de software pueden permanecer estables durante toda la vida esperada del software; otros pueden ser más dependientes del feedback que se obtenga del diseño o posibles cambios en la especificación inicial del cliente recogida en los requisitos de usuario. Tales requisitos inestables se deben señalar, para prestarles la atención que requieren a lo largo del desarrollo del proyecto.

- **Fuente**: se indicará de qué requisitos de usuario se deriva cada requisito software, o bien se realizarán referencias a otros sistemas, documentos o sitios de interés de los que se obtuvo información para la especificación del requisito. La fuente de un requisito no funcional puede ser así mismo una necesidad propia de un sistema con las características del presente proyecto (por ejemplo necesidad de seguridad en el sistema).

- **Verificabilidad**: cada requisito de software será verificable. Esto significa que debe ser posible comprobar fehacientemente que el requisito se ha incorporado en el diseño, es decir, que se puede probar que el software aplica el requisito.

2.- Requisitos

2.1 REQUISITOS FUNCIONALES

2.1.1. Requisitos de operación

IDENTIFICADOR: R-OP001	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta. Durante toda la vida del sistema.
DESCRIPCIÓN:	El interfaz será claro y facilmente operable, sin necesidad de aprender comandos.

2.1.2. Requisitos de información

IDENTIFICADOR: R-I001	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta. Durante toda la vida del sistema.
DESCRIPCIÓN:	Las capturas de tráfico se almacenarán en el formato estándar PCAP 2.4.

IDENTIFICADOR: R-I002	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta. Durante toda la vida del sistema.
DESCRIPCIÓN:	La información tratada y decodificada se enviará en formato de texto comprimida o no con un formato estándar de código abierto.

IDENTIFICADOR: R-I003	
PRIORIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, se podrán añadir o eliminar protocolos durante la vida del sistema.
DESCRIPCIÓN:	El sistema decodificará los protocolos DNS, HTTP, SMTP, POP, IMAP, SIP y FTP.

IDENTIFICADOR: R-I004	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☐ ESENCIAL ☑ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, se podrán añadir o eliminar protocolos durante la vida del sistema.
DESCRIPCIÓN:	El sistema decodificará los protocolos MSN, Gtalk y Yahoo..

Identificador: R-I005	
Prioridad: ☑ Alta ☐ Media ☐ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Media, se podrán añadir o eliminar protocolos durante la vida del sistema.
Descripción:	Se utilizará una base de datos para obtener la información gráfica por IP (GeoIP).

2.1.3. Requisitos de restricción a las capacidades del sistema

Identificador: R-RC001	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Equipo de Análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	No puede existir tráfico entre el dispositivo y el servidor de almacenamiento que viaje sin cifrar.

Identificador: R-RC002	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Baja, en ampliaciones del sistema la información podrá llegar en otros formatos, como PPP.
Descripción:	El sistema no atenderá a tráfico que no le venga en formato IP.

2.1.4. Requisitos de rendimiento

IDENTIFICADOR: R-REN001	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Una alarma no puede tardar más de 60” en llegar al operador desde que son generadas, y debe haber constancia de que el Operador la ha recibido.

IDENTIFICADOR: R-REN002	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, en futuras ampliaciones la capacidad del canal de tráfico puede aumentar.
DESCRIPCIÓN:	El tráfico analizable podrá ser de hasta 100MB/seg y debe ser procesado (decodificado y analizado) en no más de 60” desde que llega al interfaz de la máquina. Deseable en tiempo real.

IDENTIFICADOR: R-REN003	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
Estabilidad:	Media, debido a futuras ampliaciones del sistema, esta exigencia se podría relajar.
Descripción:	Desde el arranque de la máquina sniffer hasta que empiece a funcionar a pleno rendimiento no deben pasar más de 60”.

IDENTIFICADOR: R-REN004	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	El sistema debe guardar el tráfico de por lo menos las últimas 24h.

IDENTIFICADOR: R-REN005	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: EQUIPO DE ANÁLISIS
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	El tráfico analizado (tanto decodificado como en formato bruto PCAP) debe estar en el la máquina del Operador en menos de 1 hora desde que llegó al PC_analizador.

IDENTIFICADOR: R-REN006	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	El operador debe poseer al menos las capturas completas de la última semana.

2.1.5. Requisitos de interfaz

IDENTIFICADOR: R-INT001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se dispondrá tanto de un interfaz web como de un interfaz por consola para la interacción del Operador con el sistema Sniffer.

2.1.6. Requisitos de recursos

IDENTIFICADOR: R-REC001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: EQUIPO DE ANÁLISIS
NECESIDAD: ☐ ESENCIAL ☐ DESEABLE ☑ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, podrán variar las fechas en función de la evolución del proyecto.
DESCRIPCIÓN:	La fecha de entrega del primer prototipo se fija en 6 semanas desde el inicio del proyecto.

IDENTIFICADOR: R-REC002	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: EQUIPO DE ANÁLISIS
NECESIDAD: ☐ ESENCIAL ☑ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, variará en función de los factores de riesgo.
DESCRIPCIÓN:	La fecha de entrega del primer prototipo funcional se fija en entre 12 y 16 semanas desde el inicio del proyecto.

2.1.7. Requisitos de verificación

IDENTIFICADOR: R-VER001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: INDECT
NECESIDAD: ☐ ESENCIAL ☐ DESEABLE ☑ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se aislará y testeará en laboratorio el sistema, una vez finalizado y entregado, para verificar la autenticidad de sus resultados sin conexiones a sistemas externos.

2.1.8. Requisitos de pruebas de aceptación

Identificador: R-PA001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se probarán los protocolos HTTP, FTP, SMTP, POP3, IMAP y FTP, transmitiendo aleatoriamente palabras sospechosas por ellos buscando la generación de alertas.

2.1.9. Requisitos de documentación

IDENTIFICADOR: R-DOC001	
PRIORIDAD: ☐ ALTA ☐MEDIA ☑ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se documentará todo el código para hacerlo facilmente mantenible.

IDENTIFICADOR: R-DOC002	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: INDECT
NECESIDAD: ☐ ESENCIAL ☑ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, variará en futuras ampliaciones del sistema.
DESCRIPCIÓN:	Se realizará un documento (anexo) explicando brevemente, entre 5 y 10 páginas, la complejidad del sistema, funcionalidades y diseño del mismo.

2.1.10. Requisitos de seguridad del sistema frente a amenazas

IDENTIFICADOR: R-SSA001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se buscará un hardware de tamaño pequeño y robusto para facilitar el camuflaje del sistema en unas instalaciones de acceso público.

IDENTIFICADOR: R-SSA002	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se utilizará un color discreto para el hardware, de cara a facilitar su camuflaje en unas instalaciones de acceso público.

IDENTIFICADOR: R-SSA003	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se buscará un hardware cuyas emisiones acústicas sean inferiores a 10 dB de cara a facilitar su camuflaje en unas instalaciones de acceso público.

IDENTIFICADOR: R-SSA004	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se debe valorar en la elección del hardware aquel cuyo nivel de emisiones radioeléctricas sea el menor posible.

IDENTIFICADOR: R-SSA005	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Será necesario un sistema de alimentación eléctrico auxiliar preveyendo que no halla tomas eléctricas en el lugar de instalación del sniffer.

Identificador: R-SSA006	
Prioridad: ☐ Alta ☐ Media ☑ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☐ Alta ☑ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	El sistema debe seguir inmune a interferencias radioeléctricas.

Identificador: R-SSA007	
Prioridad: ☐ Alta ☐ Media ☑ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☐ Alta ☑ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	A nivel software, como medida de seguridad ante un posible secuestro o contraespionaje del hardware instalado, es necesario disponer de medidas de seguridad para no facilitar una disección del sistema.

2.1.11. **Requisitos de portabilidad**

Identificador: R-POR001	
Prioridad: ☐ Alta ☐ Media ☑ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	El sistema debe funcionar entre temperaturas de 0ºC y 40ºC

Identificador: R-POR002	
Prioridad: ☐ Alta ☐ Media ☑ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☐ Alta ☑ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	El sistema debe funcionar en ambientes con una humedad alta (por determinar en posteriores revisiones del documento).

IDENTIFICADOR: R-POR003	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: INDECT
NECESIDAD: ☐ ESENCIAL ☐ DESEABLE ☑ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, en futuras ampliaciones se podrán intervenir líneas de otro medio físico, como fibra óptica.
DESCRIPCIÓN:	La parte del sniffer debe ser instalable en una línea telefónica en menos de 4 minutos desde que se localiza la línea.

IDENTIFICADOR: R-POR004	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: EQUIPO DE ANÁLISIS
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	El sistema completo debe ser transportable en una maleta de operario de servicio telefónico.

2.1.12. Requisitos de prevención de errores del usuario

IDENTIFICADOR: R-PEU001	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: EQUIPO DE ANÁLISIS
NECESIDAD: ☐ ESENCIAL ☐ DESEABLE ☑ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se grabarán todos los comandos ejecutados en consola por el operador en el PCSniffer en un fichero log.

2.1.13. Requisitos de manejo de errores

IDENTIFICADOR: R-ME001	
PRIORIDAD: ☐ ALTA ☐ MEDIA ☑ BAJA	FUENTE: INDECT
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	Se implementará un sistema de autodiagnóstico que genere avisos al operador.

Identificador: R-ME002	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	Ante la previsión automática de colapso del sistema, se provocará un reinicio previo aviso (no autorización) al operador.

2.1.14. Requisitos de mantenimiento

Identificador: R-MAN001	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	Se debe proveer de todo el código fuente generado.

Identificador: R-MAN002	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	Se debe generar la documentación adecuada para facilitar el mantenimiento del sistema.

Identificador: R-MAN003	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Equipo de análisis**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	El sistema debe llevar al menos un puerto de ampliación disponible para futuras funcionalidades.

Identificador: R-MAN004	
Prioridad: ☐ Alta ☑ Media ☐ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	Como punto crítico y fundamental se utilizará, tanto en el hardware como en el software, la última tecnología disponible en el mercado, de cara a crear un producto que perdure el máximo tiempo posible, sea fácilmente mantenible y no haya problemas de abastecimiento en un futuro.

2.1.15. Requisitos de seguridad de las personas frente a fallos

Identificador: R-SPF001	
Prioridad: ☑ Alta ☐ Media ☐ Baja	**Fuente: Indect**
Necesidad: ☑ Esencial ☐ Deseable ☐ Opcional	
Verificabilidad: ☑ Alta ☐ Media ☐ Baja	
Estabilidad:	Alta, durante toda la vida del sistema.
Descripción:	Las comunicaciones del usuario monitorizado no deben ser alteradas en ningún aspecto.

2.2 REQUISITOS INVERSOS

Cabe destacar que para los requisitos inversos no se indicarán valores para su prioridad, pues no llegarán a implementarse, ni para su necesidad, pues no puede decirse en qué grado son o no necesarios, ya que recogen funcionalidades que no ofrecerá el sistema.

Por otra parte, resulta importante recordar que se pretende con estos requisitos inversos recoger y aclarar únicamente aquellas circunstancias que pueden producir confusión en cuanto a cuáles serán las funcionalidades proporcionadas o no.

IDENTIFICADOR: R-INV001	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: INDECT**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Alta, durante toda la vida del sistema.
DESCRIPCIÓN:	No se puede utilizar software que no utilice licencia GNU v2 o compatible.

IDENTIFICADOR: R-INV002	
PRIORIDAD: ☐ ALTA ☑ MEDIA ☐ BAJA	**FUENTE: EQUIPO DE ANÁLISIS**
NECESIDAD: ☑ ESENCIAL ☐ DESEABLE ☐ OPCIONAL	
VERIFICABILIDAD: ☑ ALTA ☐ MEDIA ☐ BAJA	
ESTABILIDAD:	Media, en un futuro se pueden realizar ampliaciones para interactuar con otros tipos de líneas.
DESCRIPCIÓN:	No interceptará líneas de comunicaciones que no sean DSL.

Anexo IV: "Tecnologías aplicables"

Descripción de las alternativas para cada componente planteado en el diseño del sistema.

[c001] Interceptor_DSL

[c001.001] Acoplado a la línea.

[c001.002] Generador de señal.

El diseño de este componente queda excluido de este documento.

[c002] Router

C002.001 Hardware

C002.001.001 Router: debe ser compatible con **C001.001.002.**

Para cubrir las necesidades de este sistema se puede realizar de 3 maneras distintas:

- Router.
- Modem ADSL USB
- Modem ADSL Minipci

C002.002 Software

C002.002.001 Firmware: Entre los firmwares GNU, con soporte constante por parte de la comunidad desarrolladora, con rendimiento contrastado y modificables para conseguir las especificaciones concretas de este proyecto destacan:

http://openwrt.org/
http://es.wikipedia.org/wiki/DD-WRT

El diseño de este componente queda excluido de este documento.

[c003] PCSniffer

C003.001 HARDWARE

[c003.001.001] Microprocesador: en el mercado existe gran variedad de microprocesadores:

65xx	LatticeMico32	Nios	AVR	Motorola	SuperH family	PA-RISC	Itanium
M32R	X86, X86-64	MIPS	EISC	IBM POWER	NSC 320xx	OpenRISC	DEC
scamp	Signetics 2650	SPARC	RCA	Transmeta	INMOS	ARM	XAP

No obstante, al condicionar la elección del procesador a la del sistema operativo (software) y viceversa, y estimándose como se verá en el componente **C003.002.001** que se utilizará un sistema operativo GNU/Linux basado en Debian, según la tabla de compatibilidades de éste se pueden eliminar tipos de procesadores hasta quedarse con esta selección:

Arquitectura	Designación de Debian	Subarquitectura	Sabor
Basada en Intel x86	i386		
AMD64 e Intel EM64T	amd64		
DEC Alpha	alpha		
ARM	arm	Netwinder y CATS	netwinder
	armel	Versatile	versatile
	arm and armel	Intel IOP32x	iop32x
		Intel IXP4xx	ixp4xx
		Marvell Orion	orion5x
HP PA-RISC	hppa	PA-RISC 1.1	32
		PA-RISC 2.0	64
Intel IA-64	ia64		
MIPS (big endian)	mips	SGI IP22 (Indy/Indigo 2)	r4k-ip22
		SGI IP32 (O2)	r5k-ip32
		MIPS Malta (32 bit)	4kc-malta
		MIPS Malta (64 bit)	5kc-malta
		Broadcom BCM91250A (SWARM)	sb1-bcm91250a
		Broadcom BCM91480B (BigSur)	sb1a-bcm91480b
MIPS (little endian)	mipsel	Cobalt	cobalt
		MIPS Malta (32 bit)	r4k-malta
		MIPS Malta (64 bit)	5kc-malta
		Broadcom BCM91250A (SWARM)	sb1-bcm91250a
		Broadcom BCM91480B (BigSur)	sb1a-bcm91480b
IBM/Motorola PowerPC	powerpc	PowerMac	pmac
		PReP	prep
Sun SPARC	sparc	sun4u	sparc64
		sun4v	
IBM S/390	s390	IPL del lector VM y DASD	genérico
		IPL de cinta	cinta

De este modo el rango de elección se acota ligeramente. Así mismo, entendiendo que el hardware ha de ser una arquitectura estándar, totalmente conocida y probada, con gran cantidad de soporte y empresas que lo respalden, se decide acotar la selección a los siguientes 4 tipos de procesadores en base a su alta disponibilidad en el mercado:

PROCESADOR	DESCRIPCIÓN
Intel x86	Familia de procesadores Intel de 32 bits.
Intel IA64	Gama Intel de procesadores de 64 bits.
AMD64	Gama AMD de procesadoresde 64 bits compatible con IA64.
ARM	Familia ARM

Por lo que se analizará cada microprocesador:

ARM
Ventajas • Bajo consumo eléctrico. • Es el procesador más vendido (80%), presente en teléfonos móviles, electrónica, etc. • Versión de 64 bits disponible.
Inconvenientes • La mayor parte de los entornos de desarrollo y librerías para este procesador son comerciales, no libres.

INTEL X86 (32 BITS)
Ventajas: • Es el procesador universal más cónocido en el desarrollo de aplicaciones informáticas. Su arquitectura es utilizada en satélites. • Ofrece un rendimiento medio. • Alta disponibilidad.
Inconvenientes: • La arquitectura de 32 bits, si bien es la predominante actualmente, está en receso de cara a la migración a arquitecturas más modernas y potentes. • Alto consumo de energía. • Limitaciones hardware: no puede direccionar más de 4GB de RAM.

INTEL IA64, AMD64
Ventajas: • Pueden direccionar hasta 128GB de RAM, lo cual facilita la escalabilidad de cualquier producto basado en esta arquitectura.
Inconvenientes: • Compatibles con aplicaciones diseñadas para arquitecturas Intel x86-32 bits. • Es la línea actual de procesadores para ordenadores. • Mayor uso de memoria en los direccionamientos (64 bits) y uso de caché.

Así, la opción de ARM queda descartada por su escasez de recursos de software libre, lo cual a medio plazo la hace poco viable. A su vez la arquitectura x86 se descarta tanto por su limitación de 4GB de memoria RAM como por el progresivo proceso que está sustituyéndola por procesadores de 64 bits, a parte del alto consumo de energía.

Por todo ello, se estima la mejor opción un **procesador AMD ó Intel de 64 bits**, el cual cubre todas las necesidades técnicas, garantiza abastecimiento y es compatible con la mayoría de sistemas operativos y plataformas de desarrollo, así como

posibilita manejar más de 4GB de memoria RAM, algo bastante probable en este sistema debido la necesidad de obtener resultados en tiempo real.

C003.001.002 Placa base:

- *Condicionada por* **c003.001.001 Microprocesador***: AMD64 ó Intel IA64.*
- *Condicionada por* **C003.001.004, Puerto de conexión auxiliar.**
- *Condicionada por* **C003.001.005, Interfaz de red de entrada.**
- *Condicionada por* **C003.001.006, Interfaz de red para conexión con operador***.*

Así mismo el tamaño ha de ser reducido según el requisito R-SSA001 para facilitar la discrección y despliegue de este componente en producción, por lo que se evaluarán la gamas “Mini-ITX” , “Pico-ITX” ó “Micro-ITX”.

Además, y basándose en el deseo de este proyecto de hacerlo 100% con software libre, se puede estudiar un subconjunto de las gamas anteriores que llevan BIOS compatible con Linux; la tecnología **Coreboot** posibilita que el arranque de la máquina sea un 66%, en general, más rápido que el habitual. (requisito R-REN003).

CoreBoot es un proyecto que se conocía con el nombre de LinuxBIOS, siendo de software libre, respaldado por la Free Software Foundation, dirigido a reemplazar el firmware no libre de los BIOS propietarios encontrados en la mayoría de los computadores, por una BIOS ligera diseñada para realizar solamente el mínimo de tareas necesarias para cargar y correr un moderno sistema operativo de 32 bits. Coreboot actualmente tiene soporte para arquitecturas de 64bits en Coreboot v2.

C003.001.003, Almacenamiento:

Teniendo en cuenta que la capacidad mínima calculada debe ser de 800GB, la solución se podría diseñar de dos maneras:

a) **Tarjeta externa**. El PC_sniffer dispondría de una ranura con una tarjeta tipo “Compact Flash”

El SO estaría instalando en un HD interno (SATA II ó USB) y la partición con las copias de seguridad y datos temporales en otro disco. De cara a la recolección de los resultados, si se quiere evitar que estos viajen por canales inseguros, se posibilita el reemplazo diario del disco por otra evitando cualquier problema de seguridad de los mismos.

Actualmente no hay tarjetas para cubrir las necesidades de espacio de este componente, 800GB.

b) **Disco duro industrial**:

Modelo	Características	Capacidad	Vel. Lectura	Vel. Escritura	Precio/GB	Coste
Buffalo Solid-State Drives 2,5" 30 GB	Negro, SHD-NSUM30G	30,0 GB	150 MB/seg	90 MB/seg	3,23 €	96,99 €
Transcend TS32GSSD25		32,0 GB	150 MB/seg	90 MB/seg	3,53 €	112,99 €
Patriot Solid-State Disk 2,5" 32 GB	Negro, Warp Series	32,0 GB	175 MB/seg	100 MB/seg	3,59 €	114,99 €
OCZ Solid-State Drive 2,5" 30 GB	Negro, OCZSSD2-1SLD30G, Solid Series	30,0 GB	155 MB/seg	90 MB/seg	4,23 €	126,90 €

Los discos duros industriales Solid State Drive son también una buena opción. Su resistencia en ambientes industriales es mayor que le los discos habituales, así como su rendimiento en ambientes distintos a los habituales. Además disponen de más capacidad y mayor velocidad que los discos flash, pero su precio es consecuentemente mayor. Se descarta el formato IDE a favor del SATA, estándar actual. Los candidatos son:

Siendo 800 GB una capacidad suficiente para este proyecto, el segundo criterio de elección es la **velocidad de acceso.** Por lo tanto, la opción más interesante utilizando esta tecnología sería:

C003.001.004, Puerto de conexión auxiliar

Igual que en los componentes anteriores, se necesita una tecnología ámpliamente usada y estándar de la industria.

Los requisitos para esta conexión son:

- No quedar desfasado en 5 años.
- Disponer de un ancho de banda de 10MB/seg al menos.
- Facilmente mantenible, documentado y apoyado por la industria.

Se presentan por tanto varias alternativas para la transmisión de datos genéricos:

PUERTO	Año de creación	Velocidad
Puerto serie	1969	0'020 KB/seg
Puerto paralelo	1981	0’150 MB/seg
SCSI	1986	2 MB/seg
RJ45 cat 6	1995	1000MB/seg
USB 2.0	2000	40 MB/seg
IEEE 1394b Firewire 800	2002	320 MB/seg

Los puertos serie, paralelo y SCSI se encuentran ya en desuso y su fabricación es excepcional. Esto supondría un riesgo a la hora de producir la máquina en cadena.

Sobre el puerto RJ45, su uso implica parametrizar la conexión entre ambos dispositivos como si fuese una red, definiendo un protocolo de red (como IP) y otro de transmisión (como TCP), a parte de un protocolo específico para el diálogo entre ambos elementos. Además, ya se dispone de conexiones RJ45, por lo que la elección de cualquier otro tipo de conector aumenta la escalabilidad y opciones de este sistema.

El conector Fireware es técnicamente la mejor elección, ya que goza del mayor ancho de banda y un diseño moderno, no obstante su aceptación por el mercado no es la que teóricamente se podría pensar, y esto es un hándicap que devalúa su candidatura a ser la interface a utilizar en ese módulo, ya que es necesario utilizar el conector en voga más utilizado y que menos problemas de mantenimiento o logística pueda producir.

Respecto al interfaz USB, cabe destacar el ancho de banda que cubre las necesidades de este proyecto sobradamente, la naturaleza de estándar abierto cumpliendo las perspectivas del proyecto y la sobrada aceptación de este conector, actualmente el más utilizado en el mundo.

Un factor más a la hora de evaluar el uso del USB son las directrices de la UE, en las que por ejemplo implicarán que en 2010 todos los cargadores de teléfonos móviles utilicen el estándar USB, lo cual afianzará esta tecnología sobradamente como estándar universal de comunicación de datos entre periféricos, aparte de abaratar la producción de este conector. Esto hace que el uso de USB sea previsto con el estandar de facto en la interconexión ligera de dispositivos [3]

En definitiva, la **elección del USB** cubre sobradamente las necesidades tecnológicas requeridas y es la mejor elección para los aspectos de mantenimiento, producción y administrativos de este proyecto.

C003.001.005, Interfaz de red de entrada:

Partiendo de un tráfico máximo de 100 **MB**/seg (requisito R-REN002), el PC_sniffer se conectará al componente Router (C002). Será necesario utilizar una interfaz de red Ethernet, entre las que se barajan las siguientes aternativas:

Tecnología	Velocidad
Ethernet 10/100	100MB/seg
Ethernet 10/100/1000	1000MB/seg

Siendo el tráfico de entrada previsto de 100MB/seg, la interfaz de red incorporada por defecto en el componente “c003.001.001 Placa base” cumple con los requisitos propuestos. El modelo Realtek RTL8168C dispone de las características requeridas y dispone de driver en GNU/Linux.

3 http://www.aecomo.org/content.asp?contentid=10843&contenttypeid=2&catid=267&cattypeid=2

C003.001.006, Interfaz de red para conexión con operador.

Por este canal viajarán 3 tipos de información:

- Alertas y tráfico de gestión.
- Impresión automática de alertas en la impresora.
- Envío de las capturas, que en una estimación fatalista sería:

UPLOAD MÍNIMO NECESARIO		
Máximo tráfico generado en un día por un usuario en una DSL de 100MB	518'4GB	518'4GB
Compresión de este tráfico con pigz (33%)	518'4GB * 0'33	171'07 GB
Envío del tráfico de un día en 24h y compreso.	171'07GB/24h	7'128GB/h
Conversión a MB/seg	7'128GB/h*1000MB /3600seg	**1'98MB/seg ó 19'8 Mbps**

Para las necesidades de este componente se barajan las siguientes soluciones:

Tecnología	Velocidad
Modem 3G	Download 12 Mbps Upload 2'1 Mbps
Wifi	54MB/seg
Ethernet	10/100/1000 MB/seg

Modem 3g
Ventajas
- Rápido despliegue del dispositivo. - Comunicación asegurada, radioespectro reservado.* * No se valoran medidas de contraespionaje.
Inconvenientes
- Dependencia de una compañía privada para el funcionamiento del sistema. - Ancho de banda de subida insuficiente para la retransmisión del tráfico en tiempo real, 2'1Mbps frente a 19'8Mbps necesarios. - Coste por giga transmitido.

Wifi
Ventajas
- Fácil despliegue. - Ancho de banda suficiente para cualquier situación. - Comunicación asegurada, cifrados de alto nivel disponibles.
Inconvenientes
- Se desvela facilemente la existencia de una nueva red inalámbrica, problema en zonas despobladas.

Ethernet
Ventajas
- Gran ancho de banda (hasta 1000mbps) - Rendimiento y fiabilidad de la conexión altos al ser un medio no alterable.
Inconvenientes
- Instalación engorrosa y poco discreta al ser un cable de un diámetro distinto al habitual telefónico.

Tanto en 3G como en Wifi se recomienda el uso de una **antena direccional**, para mejorar el ancho de banda y rendimiento del enlace, de modo que se cubriría el requisito R-SSA006 sobre interferencias intencionadas.

Con la tecnología 3G, se contrataría con un ISP un acceso de 12MB se cubriría el caudal de bajada pero no el de subida. Además implicaría un coste relativamente elevado, y la conexión podría correr el riesgo de ser intermitente o con fluctuaciones de velocidad dependiendo del operador. El envío automático de las capturas se desactivaría, descargando el operador manualmente, a través de la web, las descargas que desease.

Por último, con una conexión Ethernet se necesitaría desplegar un cable RJ45 conectado con el equipo del Operador, lo cual es complicado. Mientras, si se utilizase una conexión wifi con cifrado alto y cambio frecuente de clave, se dispondría de un ancho de banda suficiente, pero con el handicap de tener que localizar el PCOperador en un radio relativamente corto y evaluar que se estaría emitiendo señales de radio, lo cual reduce la discrección del sistema.

Por tanto, para este punto se proponen las 3 tecnologías, dejando a elección del Operador cual utilizar en cada escenario.

C003.001.006, Interfaz de red de backup.

La segunda interfaz de red, utilizada de backup de la primera en caso de que falle, será el mismo modelo decidido en el **C003.001.005** en formato PCI-Express o, de no ser posible, en USB. Al ser el mismo modelo, se simplifica la instalación y necesidad de drivers.

C003.001.007, RAM

Según el componente “C003.xxxx Placa base”, la memoria RAM compatible es del tipo DDR3 y se disponen de 4 slots de memoria.

Por ello, se **barajan** las siguientes opciones:

Memoria	Cantidad	TOTAL
DDR3 24.576 MB 1600Mhz	2	47.152 MB
DDR3 24.576 MB 1600Mhz	4	94.304 MB

C003.001.008, Alimentación eléctrica

Para evitar problemas eléctricos, tanto de suministro como picos de tensión, se implantará un SAI. Por ello se estudia previamente el consumo del dispositivo:

Dispositivo	Consumo/hora
Ej: Asus EEE PC 901	12 W
Ej: Placa Mini ITX	15 W

No obstante estos consumos son estándares, sin hacer uso alto de CPU y wifi. Mientras, un SAI como el Powerware 5125 1500VA dispone de hasta 1050 vatios, por lo que en el caso, por ejemplo, de un Asus EEE funcionando a pleno rendimiento podría mantenerlo activo el siguiente número de horas:

SAIPowerware(1050w/h) /ConsumoAsus(12w+5w CPU+5w wifi) = **47h***

*No se incluyen las horas de energía que además podría proporcionar la batería del propio equipo.

[c003.002] SOFTWARE

C003.002.001, Sistema operativo:

Según los requisitos, al menos se deben cumplir 3 condiciones:

Sist. Operat.	GNU	Fiablidad y estabilidad	Soporte multiprocesador
Windows	No	N/A*	Si
MAC	No	N/A*	Si
Ubuntu	Si	Si	Si
Debian	Si	Si	Si
Fedora	Si	Si	Si
SuSE	Si	Si	Si
Plan9	Si	Si	Si
FreeBSD	Si**	Si	Si
OpenBSD	Si**	Si	Si
Solaris	No	N/A*	Si

**: el soporte depende exclusivamente de la empresa comercializadora, lo cual puede generar demoras o no realización del mismo en ciertas situaciones. El no ser software GNU causa que no cumpla el requisito XXXX de este documento, lo cual descarta automaticamente este software en este proyecto.*

***: la licencia BSD cubre la licencia GNU y no genera incompatibilidades en este proyecto.*

Sistema operativo	Debian GNU/Linux	Fedora (Linux)	SUSE Linux	FreeBSD	OpenBSD	Plan 9
Creador	Proyecto Debian	Proyecto Fedora	SuSE	Universidad de California	Theo de Raadt	Bell Labs
Año de primera distribución	1993	2003	1994	1993	1996	1993
Aspectos generales						
Última versión estable	5.0 Lenny	11	39824	7	36617	Fourth Edition
Costo	Gratuito	Gratuito	Gratuito	Gratuito	Gratuito	Gratuito
Licencia	Libre: GNU	Libre: GNU	Libre: GNU	Libre: BSD	Libre: BSD	Libre: LPL
Tipo de usuario	Hogar, ciencia, servidores, redes, negocios	Hogar, ciencia, servidores	Hogar, ciencia, servidores	Servidores	Servidores	Estaciones de trabajo, servidores, embebido HPC
Aspectos técnicos						
Tipo de núcleo	Monolítico	Monolítico	Monolítico	Monolítico	Monolítico	Monolítico
Arquitecturas de procesador soportadas	Intel x86, Intel IA64, AMD64, DEC Alpha, ARM, HP PA-RISC, MIPS (big endian), MIPS (little endian), PowerPC, IMB S/390, Sparc	Intel x86, AMD64, PowerPC	Intel x86, AMD64, PowerPC	Intel x86, Intel IA64, AMD64, DEC Alpha, ARM, MIPS, pc98 (NEC PC-98x1), PowerPC, UltraSparc, Sun4v (UltraSparc-T1), XBox (Microsoft X-Box)	Intel x86, AMD64, DEC Alpha, ARMISH, HP300, HP PA, Landisk, Luna-88k, Mac68k, PowerPC, Motorola VME 68k, Motorola VME 88k, SGI, Sparc, UltraSparc, VAX, Zaurus	Intel IA32, PowerPC, ARM, DEC Alpha, MIPS, Sparc, Motorola 68000
Sistema de archivos por defecto	ext3	ext4	ReiserFS	Berkeley FFS	Berkeley FFS	fossil/venti, 9P2000, kfs, ext2, FAT, ISO 9660
Soporte de sistemas de archivo de 16 bits	?	Sí	?	Sí	?	?
Soporte de sistemas de archivo de 32 bits	Sí	Sí	Sí	Sí	Sí	Sí
Soporte de sistemas de archivo de 64 bits	Sí	Sí	Sí	?	?	?
Herramienta de actualización por defecto	apt	yum	YaST	Fuentes	Fuentes	replica/pull
Aspectos gráficos						
Entorno gráfico[1]	Aplicación: X Window System	Aplicación: X Window System	Aplicación: X Window System	Aplicación: X Window System	Aplicación: X Window System	Aplicación: rio
Sistema de ventanas por defecto	GNOME	GNOME	KDE	?	N/A	rio

Por lo que se decide, en función de las incomaptibilidades de los demás sistemas operativos y uso por defecto ext3, centrarse en Debian y Fedora.

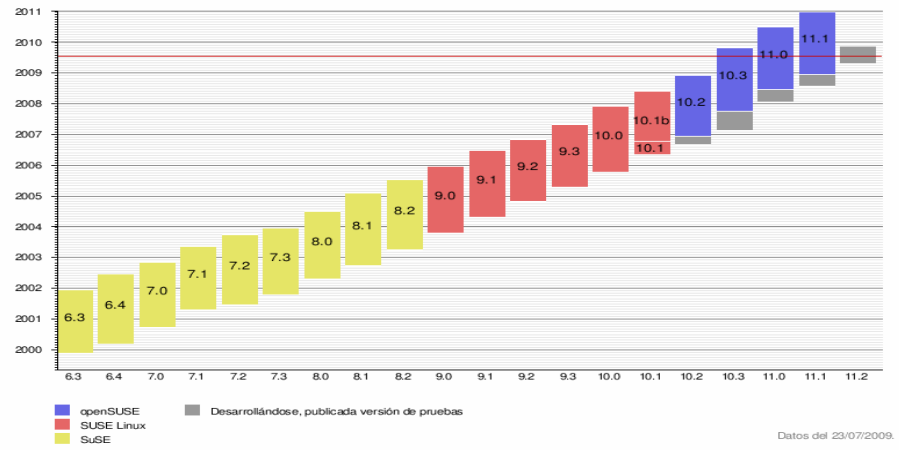
Puesto Distribución Distrowatch.com	Distribución	HPD
1º	Ubuntu*	2109
2º	Fedora	1560
4º	openSUSE	1235
6º	Debian	898

Fecha de obtención de datos: 14/10/2009

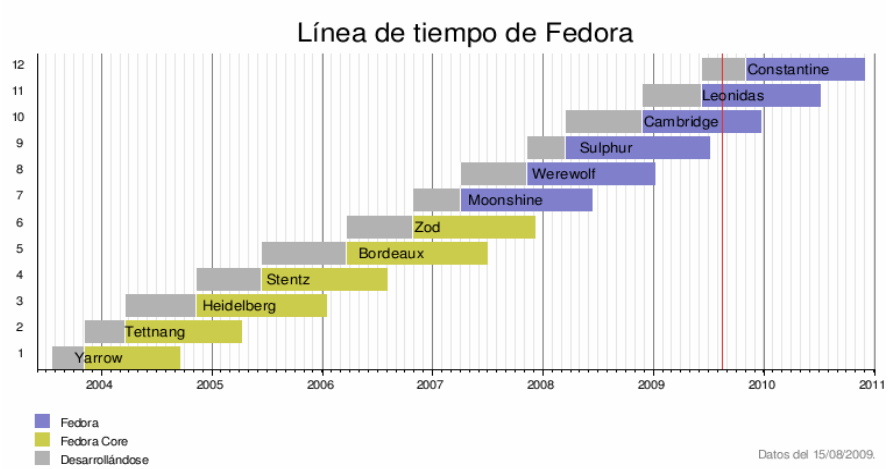
* No se incluye Ubuntu en el estudio al ser una distribución derivada de Debian.

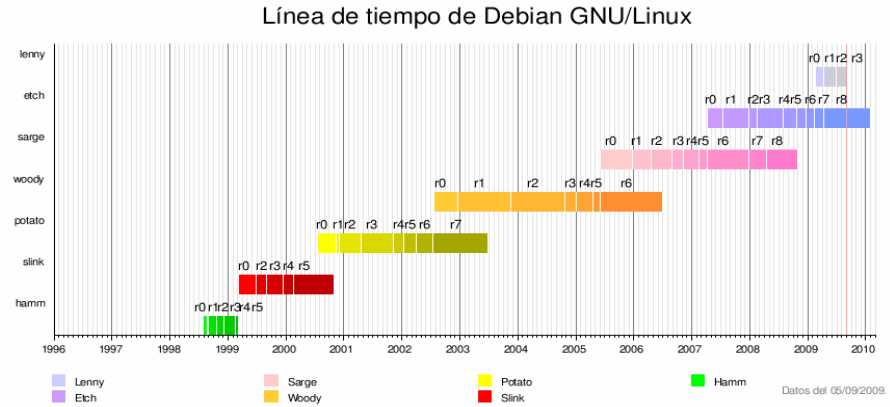
Por lo que se valora la utilización de 3 distribuciones Linux: Debian, Fedora y openSuSE. Este análisis coincide con el top 6 de Distrowatch, que muestra las distribuciones más utilizadas.

OpenSUsE



Fedora





Cronologías de las distintas ediciones de las distribuciones GNU/Linux en estudio, así como etapas de mantenimiento.

Un largo período de mantenimiento de la distribución favorece su elección para el desarrollo del sistema con ella.

- OpenSUSE: 2 años.
- Fedora: 2 años y 2 meses aprox.
- **Debian: más de 3 años.**

C003.002.002: Lenguaje de programación para el binario.

Al requerirse, al menos inicialmente, que la naturaleza de los componentes sea GNU siempre que sea posible, el framework de Microsoft (.NET) quedaría descartado. Java a su vez tampoco es GNU. Por su parte, Mono es una implementación libre y GNU, del framework de .NET,por lo que quedan **c++** y **Mono** para el análisis.

- ➔ .NET: No cumple la licencia GNU, es un lenguaje propietario, no se analizará.
- ➔ JAVA: No cumple la licencia GNU, es un lenguaje propietario, no se analizará.

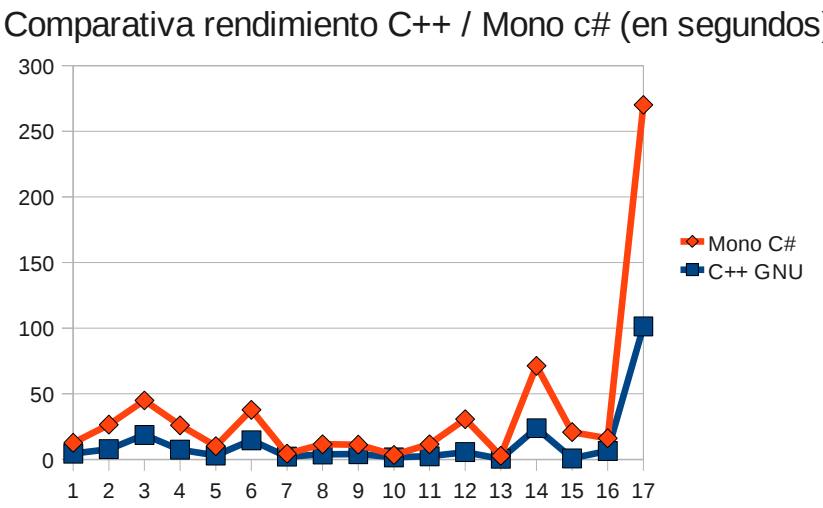
Así mismo, el requisito de multiplataforma lo cumplen los dos lenguajes, por lo que se evaluará en función de su rendimiento, al ser un sistema que ante otras necesidades premia la ejecución rápida con el objetivo de funcionar en tiempo real. Para ello se realiza una batería de pruebas codificando varios programas en ambos lenguajes, y tomando medidas tales como segundos en ejecutar el programa completo, consumo de RAM y ocupación el disco duro.

Comparativa de rendimiento “Mono c# vs GNU C++”

Programas de prueba	Segundos			Memoria (KB)			Tamaño (KB)		
	C++ GNU	Mono C#	% Diferencia	C++ GNU	Mono C#	% Diferencia	C++ GNU	Mono C#	% Diferencia
binary-trees	4.47	8.25	184.56	7	14.58	208.46	541	610	112.75
fannkuch	7.78	18.73	240.75	844	5.06	0.6	554	549	99.1
fasta	18.72	26.23	140.12	788	4.66	0.59	1248	1180	94.55
k-nucleotide	7.46	18.52	248.26	9.3	68.06	731.56	1380	1012	73.33
mandelbrot	3.02	7.15	236.75	896	4.59	0.51	1097	484	44.12
n-body	14.62	23.2	158.69	932	5.07	0.54	1705	1410	82.7
nsieve	2.08	2.28	109.62	5.76	9.6	166.62	313	341	108.95
nsieve-bits	3.86	7.7	199.48	3.32	7.13	214.96	494	363	73.48
partial-sums	4.05	7.28	179.75	852	4.7	0.55	531	455	85.69
pidigits	1.66	1.83	110.24	1.05	4.51	428.52	652	1026	157.36
recursive	2.4	9.22	384.17	1.01	5.06	501.98	566	435	76.86
regex-dna	5.58	25.11	450	12.7	124.28	978.31	1588	607	38.22
reverse-complement	0.54	2.2	407.41	13.29	27.14	204.21	810	727	89.75
spectral-norm	23.84	47.48	199.16	900	4.35	0.48	442	459	103.85
startup	0.86	20	2325.58				108	123	113.89
sum-file	6.47	9.66	149.3	852	4.76	0.56	260	198	76.15
thread-ring	101.28	168.87	166.74	2.96	11.96	403.92	626	476	76.04
TOTAL	208.69	403.71	**193.45**	6120.39	305.53	**4.99**	12915	10455	**80.95**

Fuente: Debian.org [4]

4 http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=gpp&lang2=gpp
http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=csharp&lang2=csharp



Por lo que se ve claramente cómo los programas de cálculo realizados en C++ llegan a ser casi un 200% más rápido que los realizados en Mono.

C003.002.003: Sistema de soporte remoto

Utilizando GNU/Linux como plataforma, el espectro de soluciones para acceso remoto se resume en:

- Telnet: protocolo básico de acceso remoto. No lleva cifrado. En desuso.
- SSH: protocolo de acceso remoto avanzado, con cifrado y permitiendo el envío de ficheros. Soporta exportación de escritorios.
- VNC: protocolo de acceso remoto gráfico.
- Rdesktop: protocolo de acceso remoto gráfico estándar.

Hay que tener en cuenta los siguientes aspectos:

1. La conexión debe llevar un cifrado mínimo de 2048 bits, por lo que se descarta Telnet
2. La conexión la iniciará siempre el operador.

Por lo que la única opción es el protocolo SSH, el cual proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando un cifrado robusto de 128 bits ó superior, hasta 32.768 bits.
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio del cifrado propuesto, lo cual los hacen extremamente difícil de descifrar y leer.

Los usuarios hostiles tienen a su disposición una variedad de herramientas que les permiten interceptar y redirigir el tráfico de la red para ganar acceso al sistema. En términos generales, estas amenazas se pueden catalogar del siguiente modo:

- *Intercepción de la comunicación entre dos sistemas* — En este escenario, existe un tercero en algún lugar de la red entre entidades en comunicación que hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada. Este ataque se puede montar a través del uso de un paquete sniffer — una utilidad de red muy común.

- *Personificación de un determinado host* — Con esta estrategia, un sistema interceptor finge ser el recipiente a quien está destinado un mensaje. Si funciona la estrategia, el sistema del usuario no se da cuenta del engaño y continúa la comunicación con el host incorrecto. Esto se produce con técnicas como el envenenamiento del DNS o spoofing de IP (engaño de direcciones IP) .

Ambas técnicas interceptan información potencialmente confidencial y si esta Intercepción se realiza con propósitos hostiles, el resultado puede ser catastrófico. Si se utiliza SSH para inicios de sesión de shell remota y para copiar archivos, se pueden disminuir estas amenazas a la seguridad notablemente. Esto es porque el cliente SSH y el servidor usan **firmas digitales** para verificar su identidad. Adicionalmente, toda la comunicación entre los sistemas cliente y servidor es cifrada. No servirán de nada los intentos de falsificar la identidad de cualquiera de los dos lados de la comunicación ya que cada paquete está cifrado por medio de una llave conocida sólo por el sistema local y el remoto.

C003.002.004: Sistema de ficheros y cifrado.

Utilizando GNU/Linux, y con licencia GNU, los sistemas de ficheros más extendidos con soporte para cifrado de particiones son:

ext3	XFS
ext4	ReiserFS

Por lo que se realiza el siguiente estudio comparativo de prestaciones:

File system	ext3	ext4	XFS	ReiserFS
Creator	Stephen Tweedie	various	SGI	Namesys
Year introduced	1999	2006	1994	2001
Original operating system	Linux	Linux	IRIX,Linux,FreeBSD	Linux

File system	ext3	ext4	XFS	ReiserFS 4
Windows 9x	Unknown	Unknown	Unknown	Unknown
Windows NT	with Ext2 IFS or ext2fsd	Unknown	Unknown	Unknown
Linux	Yes	Yes in kernel 2.6.28	Yes	with a kernel patch
Mac OS	Unknown	Unknown	Unknown	Unknown
Mac OS X	Partial with ext2fsx (treated as ext2)	Unknown	Unknown	Unknown
FreeBSD	Yes	No	Partial	No
BeOS	Unknown	Unknown	Unknown	Unknown
Solaris	Yes	Unknown	Unknown	Unknown
AIX	Unknown	Unknown	Unknown	Unknown
z/OS	Unknown	Unknown	Unknown	Unknown
OS/2	Unknown	Unknown	Unknown	Unknown

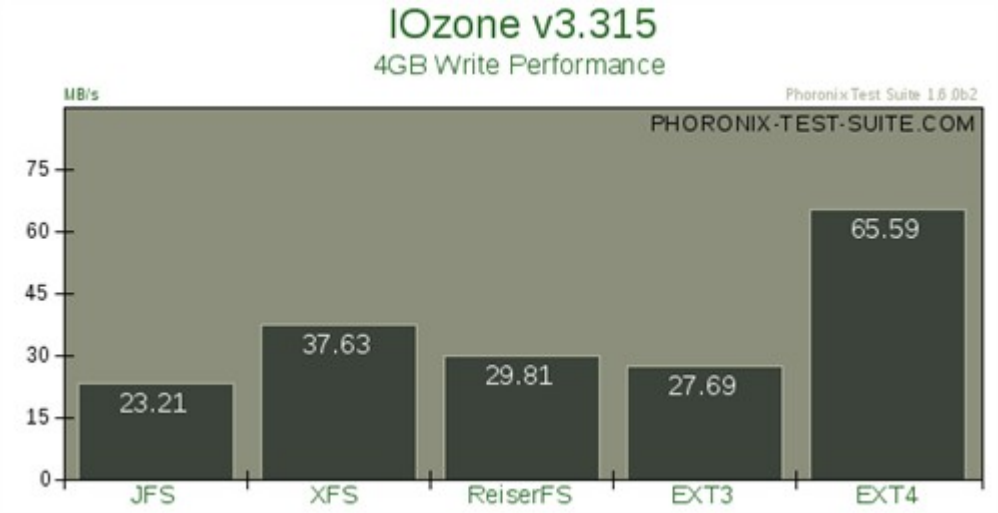
File system	ext3	ext4	XFS	ReiserFS 4
Block suballocation	No	No	No	Yes
Tail packing	No	No	No	Yes
Variable file block size [89]	No	No	No	No
Extents	No	Yes	Yes	Yes
Allocate-on-flush	No	Yes	Yes	Yes

File system	ext3	ext4	XFS	ReiserFS 4
Hard links	Yes	Yes	Yes	Yes
Symbolic links	Yes	Yes	Yes	Yes
Block journaling	Yes	Yes	No	Yes
Metadata-only journaling	Yes	Yes	Yes	No
Case-sensitive	Yes	Yes	Yes	Yes
Case-preserving	Yes	Yes	Yes	Yes
File Change Log	No	No	No	No
Snapshot	No	No	No	Unknown
XIP	Yes	Yes	Unknown	Unknown
Encryption	No	No	No	Yes
COW	No	No	Unknown	Yes
integrated LVM	No	No	No	No
Deduplication	Unknown	Unknown	Unknown	Unknown

File system	ext3	ext4	XFS	ReiserFS 4
Stores file owner	Yes	Yes	Yes	Yes
POSIX file permissions	Yes	Yes	Yes	Yes
Creation timestamps	No	Yes	No	No
Last access/ read timestamps	Yes	Yes	Yes	Yes
Last modification of content	Yes	Yes	Unknown	Unknown
This copy created	No	Unknown	Unknown	Unknown
Last metadata change timestamps	Yes	Yes	Yes	Yes
Last archive timestamps	No	No	No	No
Access control lists	Yes	Yes	Yes	No
Security/ MAC labels	Yes	Yes	Yes	No
Extended attributes/ Alternate data streams/ forks	Yes	Yes	Yes	No
Checksum/ ECC	No	Yes	No	No

File system	ext3	ext4	XFS	ReiserFS 4
Maximum filename length	255 bytes	256 bytes	255 bytes	3,976 bytes

Cabe destacar que tanto ext4, XFS como Reiser4 ofrecen un gran rendimiento general, por lo que los 3 sistemas de ficheros son factibles para este proyecto. En pruebas de rendimiento se obtiene siempre una superioridad con ext4.



Fuente: ArtsTechnica (Ene 2009).[5]

De cara a cubrir el R-SSA007 sobre cifrado del disco duro, en los sistemas de ficheros que no incorporen esta característica de manera nativa, se pueden utilizar las siguientes soluciones: TrueCrypt, el cual no tiene licencia compatible con GNU y LUKS.

Actualmente no hay estándares de cifrado de disco duro excepto Linux Unified Keys Setup ("**LUKS**"), el cual por otra parte está bien documentado y con una buena línea de soporte. Sus características más destacables son:

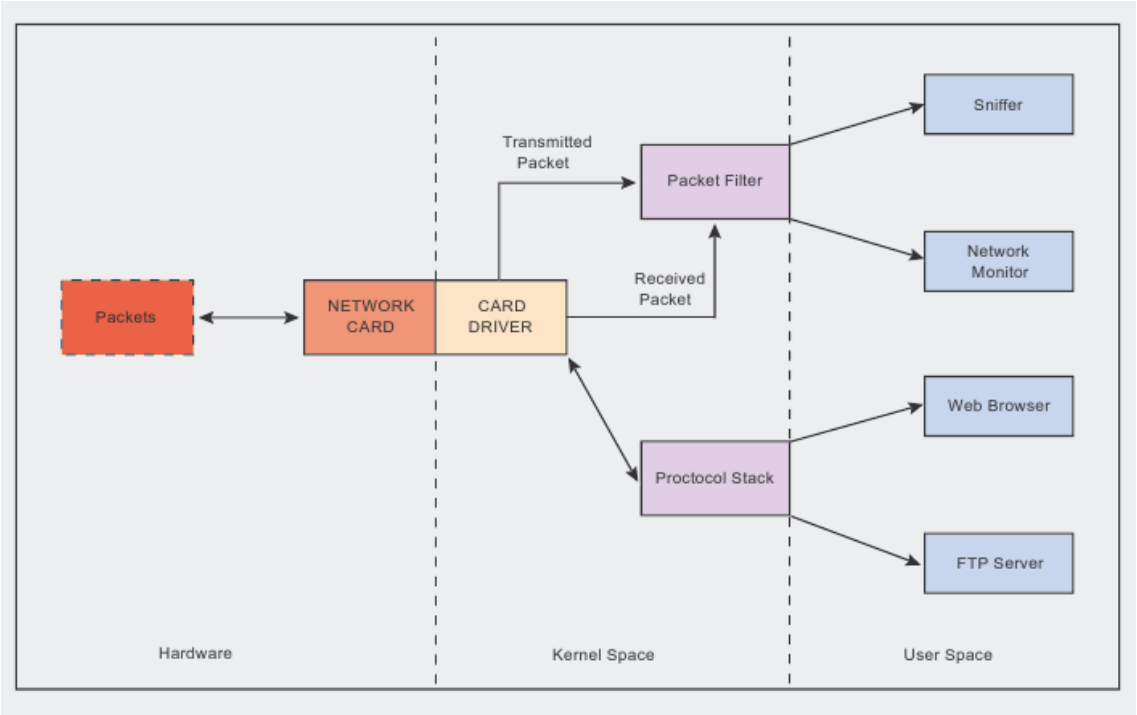
- Compatible por estandarización.
- Seguro contra ataques de baja entropía.
- secure against low entropy attacks.
- Compatibilidad con claves múltiples.
- Revocación efectiva de contraseña.
- Licencia GNU v2.

5 http://arstechnica.com/open-source/news/2009/01/super-fast-ext4-filesystem-arrives-in-ubuntu-9-04.ars

C003.002.005, Sniffer

Un sniffer es una utilidad que captura todo el tráfico de red, dirigido o no a la máquina que lo ejecuta, para posteriormente analizar esa información.

Su funcionamiento queda descrito en el siguiente gráfico:



Artículo "programación en Libpcap", Luis Martín García. [6]

de modo que su ejecución es compatible con la ejecución simultánea de aplicaciones de usuario. Al utilizar todos la librería libpcap, la fiabilidad de los más conocidos está garantizada. Entre ellos destacan:

Tcpdump	Darkstat	Ngrep	Ethereal/Wireshark/Tshark
Ettercap	Kismet	Nwatch	

No es objeto de este documento realizar una comparativa de snifers, pues daría lugar probablemente a un documento aun más amplio que el presente. Por ello, se centrará el análisis entre los 2 candidatos más utilizados, **TCPDump** y **Wireshark/Tshark**, y el nuevo sniffer **Xplico**.

C003.002.006, Decodificador.

De cara a procesar las capturas de datos, se ha de utilizar uno o varios decodificadores. Analizando la oferta existente, se pueden clasificar entre descontinuados (desarrollo abandona, sin soporte) ó incompatibles con este proyecto, y activos:

6 **http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf**

NFATs descontinuados ó incompatibles	
NetworkMiner	sólo para Microsoft Windows.
MsnShadow	Descontinuado (Setp. 2008), sólo versión para 32 bits y necesita interfaz gráfica.
Nast	Descontinuado, (Feb. 2004). Vuelca tanto cabeceras como payload en ASCII o ASCII-Hex.
NetAnalyzer	Descontinuado, (Feb. 2008).
Dsniff	Descontinuado, (2000).
Chaosreader	Descontinuado, (2003).
tcpick	Descontinuado, (2005).
aimsnarf	Descontinuado.
yahsnarf	Descontinuado.
tcpflow	Descontinuado, (2003).
nsm-console	Descontinuado, (2007).
Driftnet	Descontinuado, (2007).

A continuación se presenta una comparativa de los decodificadores de datos actuales, con los protocolos soportados.

Resumen de prestaciones de Network Forensic Analysis Tools (NFAT) en GNU/Linux – Nov. 2009

DECODIFICADOR	REQUERIDOS								DESEABLES											
	HTTP	SMTP	POP3	IMAP	DNS	FTP	SIP	MSN	GTALK	IRC	YAHOO	EMULE	TFTP	IPP	PJL	MMSE	TELNET	NNTP	FACEBOOK CHAT	RTP
Packet o Matic	✓		✓					✓		✓										
Xplico	✓	✓	✓	✓	✓	✓							✓	✓	✓	✓	✓	✓	✓	
MsnShadow								✓												
Pyflag	✓	✓	✓		✓		✓	✓		✓	✓									
tftpgrab													✓							

C003.002.007, Analizador de datos.

Este software, debido a la casuística concreta, complejidad del proyecto e integración con las demás partes, se creará desde cero. Requiere:

- **Análisis de palabras**: Las capturas de tráfico se deben analizar, palabra por palabra, en busca de palabras sospechosas.
- **API para la conexión con la BD (Sqlite3)**: la librería o wrappers para acceder a una base de datos Sqlite3 es gratuita, se utilizará la oficial de Sqlite3.

C003.002.008, compresor

Los 2 compresores habituales en GNU/Linux son Gzip y bzip2. Además, se incorporarán 7z por los buenos resultados que están aportando varias publicaciones, y XZ por la elección de la distribución Slackware como sustituto de gzip.

SOFTWARE	VERSIÓN	FECHA
Gzip	1.3.13	2/10/2009
Pigz	2.1.50	20/07/2009
Bzip2	1.0.5	10/12/2007
Pbzip2	1.0.5	8/1/2009
7z	4.65	3/2/2009
XZ	4.99	27/8/2009

Para realizar las pruebas, se utilizarán los siguientes conjuntos de datos:

INFORMACIÓN	DESCRIPCIÓN
Calgary Corpus	Conjunto de datos realizado en 1987 por Ian Witten, Tim Bell and John Clear, para testear la eficacia de compresores.
Canterbury Corpus	Conjunto de datos realizado por la universidad de Canterbury (Nueva Zelanda) en 1997 para testear la eficacia de compresores.
PCAP HTTP	Captura de tráfico exclusivamente HTTP en formato pcap v2.4.
PCAP HTTP+IM+SMTP	Captura de tráfico HTTP, IM y SMTP en formato pcap v2.4.
PCAP HTTP+IM+SMTP+P2P	Captura de tráfico HTTP, MI, SMTP y P2P en formato pcap v2.4.

Las capturas de tráfico han sido descargadas de las disponibles como muestras estándares en la web de Wireshark[7] y fundidas para proporcionar los correspondientes casos.

7 http://wiki.wireshark.org/SampleCaptures

Para determinar el tiempo, se ha utilizado el comando “*time*”

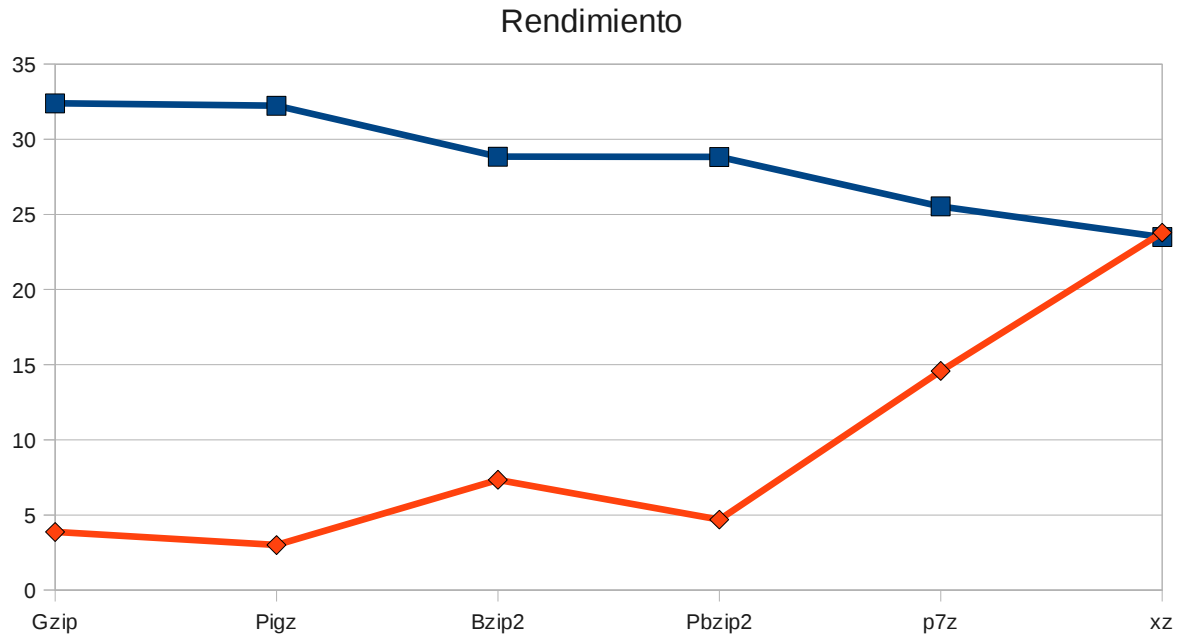
```
# time cat /etc/debian_version
lenny/sid

real    0m0.003s
```

Y como plataforma, la propuesta para el desarrollo de este sistema al inicio de este documento (apartado **C003.002.001**).

Comparativa de compresores en GNU/Linux – Oct. 2009.

Muestra de datos	Gzip		Pigz		Bzip2		Pbzip2		P7z		XZ	
	%	T	%	T	%	T	%	T	%	T	%	T
Calgary Corpus 3266560 bytes	32.72	0.8	32.67	0.57	27.08	1.2	27.16	0.94	26.3	2.1	26.22	3.52
Canterbury Corpus 2821120 bytes	26.2	0.61	26.01	0.48	20.24	1.02	20.07	0.75	17.25	2.6	17.23	3,48
PCAP HTTP 293250 bytes	29.41	0.06	29.23	0.07	26.68	0.14	26.67	0.14	19.48	0.43	19.42	0,51
PCAP HTTP+IM+SMTP 347636 bytes	28.49	0.06	28.41	0.08	26.48	0.18	26.48	0.16	19.49	0.45	19.39	0,58
PCAP HTTP+IM+SMTP+P2P 17145837 bytes	45.1	2.4	44.82	1.8	43.71	4.8	43.74	2.7	35.13	9	35.17	15,7
TOTALES	**32.39**	**3,87”**	**32.23**	**3”**	**28.84**	**7,34”**	**28.82**	**4,64”**	**25.53**	**14,58”**	**23.49**	**23,79”**

%: relación de compresión frente al original. Menos es mejor.
T: Tiempo total, en seg., empleado. Menos es mejor.



Según se puede analizar, Pigz obtiene mejores resultados frente a su clon monoproceso Gzip. Por otra parte, pbzip2 también mejora los tiempos empleados frente a bzip2, por lo que se deduce que en estas aplicaciones el uso de multiprocesador es una ventaja. Por otra parte, el uso de p7z y XZ conseguirían unos ratios de compresión mayores, nunca mejorando el 5% de los dos primeros, pero probablemente no compensarían por el alto uso de procesador de exigen.

Los tiempos empleados por 7z y xz son del orden de 3 ó incluso 5 magnitudes respecto a pigz y pbzip2, mientras que los ratios de compresión apenas mejoran en un 20% respecto a los primeros (28'8% frente a 23,4% por ej), no compensa invertir mucho procesador en una aplicación de tiempo real para obtener una mejora pequeña frente a Pigz y Pbzip2.

Por lo que entre gzip y su homónimo multiprocesador (pigz), se obtiene mejor rendimiento sobre todo con ficheros grandes (los que se utilizarán por capturar todo tráfico) con el segundo. Igual ocurre entre bzip2 y pbzip2.

Luego la decisión queda por tanto entre Pigz y pbzip2.

C003.002.009, software de envío de datos

Recordando los requisitos establecidos (envío cifrado y receptor multiplataforma), y conociendo los actuales sistemas de transferencia de ficheros más estándares, se evalúan:

Protocolo	Envío cifrado	Servidor multiplataforma
FTP		
SFTP		
SCP		

- SFTP (Secure File Transfer Program) es similar a FTP, pero aplicando una capa de seguridad a todas las transmisiones. Utiliza el cifrado de SSH, clave pública para cifrado y compresión.
- SCP (Secure CoPy) es una utilidad para copiar ficheros entre dos hosts. Utiliza SSH para autenticación y transferencia de datos, a parte de clave pública para cifrado y compresión.

La diferencia entre ambos es que SFTP es un sistema de ficheros, útil para casos en los que hay que navegar por árboles de directorios, realizar operaciones de lectura, movimientos, copias y demás. Además, emplea parte del ancho de banda en estas gestiones. **SCP** mientras, utilizando la misma capa de cifrado, se limita a enviar o recibir ficheros. Consume menos memoria y necesita menos ancho de banda.

C002.002.010: Base de datos.

Partiendo de la base de utilizar software libre, los principales SGBD libres y gratuitos son:

BASE DE DATOS	LICENCIA	VALIDEZ
PostgreSQL	Licencia BSD	
Firebird	Initial Developer's PUBLIC LICENSE Version 1.0.	
SQLite	Licencia Dominio Público	
Apache Derby	Apache License v 2.0	
MySQL	Licencia Dual *	

* La licencia GNU GPL de MySQL obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia. Si un proyecto desea incorporar MySQL en su producto pero desea distribuirlo bajo otra licencia que no sea la GNU GPL, puede adquirir, bajo pago, una licencia comercial de MySQL. En este proyecto se procura siempre que sea posible realizar todo vía licencias GNU GPL, y teniendo en cuenta que el hardware puede que no se licencie bajo esta modalidad, pero si el software, es objeto de estudio este detalle. **El pago de una licencia de MySQL queda fuera de análisis, al buscarse un coste lo menor posible e interpretarse que tanto MySQL como PostgreSQL para las exigencias de este proyecto cubren las necesidades sobradamente**.

Otro factor a tener en cuenta es la adquisición de MySQL por parte de Oracle, la mayor empresa de bases de datos del mundo actualmente, lo cual imprime cierta incertidumbre al futuro del proyecto, si bien en el peor de los casos (eliminación de la licencia GPL del producto MySQL) y debido a la comunidad de usuarios tan extensa que existe, se crearía un producto similar que garantizase su continuidad.

Una análisis técnico evaluando los puntos y características críticas de cada base de datos puede ser el disponer de soporte en las siguientes características:

	SQLite	Firebird	MySQL	PostgreSQL
Tabla temporal	✓	✓	✓	✓
Vista materializada	✓	✗	✗	✗
Árbol R-/R+	✗	✗	?	✓
Hash	✗	✗	?	✓
Expresión	✗	✗	✗	✓
Parcial	✗	✗	✗	✓
Reversa	✗	✗	✗	✗
ACID	✓	✓	✓	✓
Integridad referencial	✓	✓	✓	✓
Transacciones	✓	✓	✓	✓
Unicode	✓	✓	✓	✓
Mapa de bits	✗	✗	✗	✗
Dominio	✗	✓	✗	✓
Cursor	✗	✓	✓	✓
Trigger	✓	✓	✓	✓
Funciones 5	✗	✓	✓	✓
Procedimiento 5	✗	✓	✓	✓
Rutina externa 5	✓	✓	✓	✓
Rango	✓	✗	✗	✗
Hash	✓	✗	✗	✗
Compuesto (Rango+Hash)	✓	✗	✗	✗
Lista	✓	✗	✗	✗
✗	**10**	**11**	**10**	**7**
✓	**12**	**11**	**10**	**15**

Basándose en la popularidad en Internet y el nº de usuarios, referencias, soporte y comunidad rodeando estas tecnologías, destacan Sqlite, MySQL y PostgreSQL. Las tres ofertarían una gran seguridad al proyecto de cara a su mantenimiento y proyección de la tecnología, no quedando desfasada en tiempo.

C003.002.011, sistema de impresión

Ver apartado “C004.002.001: Sistema de impresión.”

C003.002.012: Lenguaje de programación para el interfaz.

Se consideran los principales lenguajes de programación web: ASP, Java, PHP y Ruby.

➔ ASP.NET: No cumple la licencia GNU, es un lenguaje propietario, no se analizará.

➔ JAVA: No cumple la licencia GNU, es un lenguaje propietario, no se analizará.

PHP	
CARACTERÍSTICAS	
PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).	
VENTAJAS	**INCONVENIENTES**
PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).	- Se necesita instalar un servidor web. - Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número. - La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP. - La programación orientada a objetos es aún muy deficiente para aplicaciones grandes. - Dificulta la modularización. - Dificulta la organización por capas de la aplicación.

RUBY	
CARACTERÍSTICAS	
Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla. Para los que deseen iniciarse en este lenguaje pueden encontrar un tutorial interactivo de ruby. Se encuentra también a disposición de estos usuarios un sitio con informaciones y cursos en español.	
VENTAJAS	**INCONVENIENTES**
- Existe diferencia entre mayúsculas y minúsculas. - Múltiples expresiones por líneas, separadas por punto y coma “;”. - Dispone de manejo de excepciones. - Ruby puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite. - Portátil. - Permite desarrollar soluciones a bajo Costo. - Multiplataforma	- Apenas utilizado. - Reducido soporte online.

[c004] ServidorDeAlmacenamiento:

Nota: cabe reseñar que en este elemento las restricciones hardware son distintas a las del PCSniffer, al no necesitarse un funcionamiento en tiempo real. El hardware empleado por tanto puede diferir al recomendado aquí, hay más flexibilidad. Se puede implementar fácilmente con un portátil actual.

C004.001 HARDWARE

[c004.001.001] Microprocesador

Cualquier procesador superior a un Intel Core i7 920 2. 66 Ghz ó AMD Phenom II X4 965 es válido. Deseable AMD64.

C004.001.002 Placa base

Cualquier placa base compatible con los componentes “[c004.001.001] Microprocesador” y “C004.001.003, Almacenamiento” es válida. Deseable compatibilidad con Coreboot.

C004.001.003, Almacenamiento:

Teniendo en cuenta que el requisito calculado anteriormente para este hardware es una tener una capacidad de 5035'8 GB, que representa el peor caso posible: almacenamiento mínimo para un usuario que hace un uso del 100% de su línea DSL y por ausencia de hardware potente para poder comprimirlo el flujo hay que almacenarlo en tiempo real sin compresión.

Por tanto los requisitos directos son:

- Capacidad: al menos 5035'8 GB
- Velocidad de escritura: superior a 100MB/seg.
- Deseable: portable y removible.

C004.001.004, Puerto de conexión auxiliar

Se utilizará el mismo tipo de conector que en el PCSniffer, elemento "**C003.001.006, Interfaz de red para conexión con operador.**"

C004.001.005, Interfaz de red de entrada:

Se utilizará el mismo tipo de conector que en el PCSniffer, elemento "**C003.001.006, Interfaz de red para conexión con operador.**"

C004.001.006, Interfaz de red para conexión con Internet/Intranet

El interfaz de conexión del equipo del operador con su organización o con Internet será a elección del propio operador.

C004.001.007: Teclado y ratón.

Cualquier impresora de tinta o láser actual, soportada por el componente

C004.002 SOFTWARE

C004.002.001, Sistema operativo:

Se utilizará el análisis de Sistema Operativo para el componente C003.002.001, constatando que todos los candidatos disponen de gestor gráfico, el cual es necesario para acceder cómodamente al interfaz web de la aplicación.

C004.002.002: Sistema de impresión.

En GNU/Linux existen los siguientes servidores de impresión:

- **LPD**: protocolo definido en el RFC 1179 en 1990, soporta impresoras de red. Ya en desuso y sin mantenimiento.
- LPRng: version mejorada de LPD ("next ggeneration, abandonada en 2005 y retomada a finales de 2006. La última versión, a fecha de redacción de este documento, es de Mayo de 2008.
- **PDQ**: Soporta colas de impresión BSD, por Appletalk o conexión TCP, y envío de faxes. Dispone de interfaz para servidores Novel Netware. Su implementación en las distribuciones actuales es nula. Su última versión data de 2006.
- **CUPS**: Capa de impresión implementada de manera modular para los sistemas Unix/Linux. Implementa una cola de impresión en un ordenador accesible por los demás equipos de la red. Utiliza el protocolo estándar IPP (Internet Printing Protocol) y dispone de consola de comandos.

Dispone además de un amplio soporte por parte de todos los fabricantes y sistemas operativos, y se publica bajo licencia GNU v2.

C004.002.003: Sistema de ficheros.

Al ser un equipo con información sensible, se utilizará el mismo sistema de ficheros con cifrado que se decida en el componente C003.002.004.

C004.002.004: Sistema de interfaz gráfica de usuario

En este componente se pueden evaluar los 3 principales interfaces gráficos disponibles en Debian: Gnome, KDE y XFCE.

La función de este interfaz gráfico será soportar la ejecución de un navegador web. Además, se dispondrá de recursos hardware suficientes como para poder ejecutar cualquiera de los 3 interfaces, ya que en la máquina PCOperador no se realizará ninguna labor de cómputo, apenas la recepción de capturas y la interacción con el sistema a través del mencionado navegador web. Por ello, la elección de este componente es flexible.

Los 3 son compatibles por licencia. Gnome es el interfaz más utilizado por usuarios Linux, KDE por su parte suele ser el interfaz de referencia para usuarios provenientes de entornos como Ms. Windows, y XFCE cabe destacar por utilizar muy pocos recursos.

C004.002.005: Navegador web.

Los navegadores en GNU/Linux de mayor calidad actualmente son Google Chrome, Ópera y Firefox. Los dos primeros lamentablemente pese a ser gratuitos no son de código libre, no cumpliendo el requisito R-INV001 de licencia GNU, por lo que la única opción, y recomendable además por ser el navegador (Meter alguna referencia) más utilizado en GNU/Linux, es Firefox.

C005.001.001: Impresora

Un ejemplo es la HP OfficeJet H470, impresora portátil, pequeña y fácilmente desplegable. En este elemento se puede utilizar prácticamente cualquier impresora del mercado que sea compatible con las indicaciones dadas.

1. DISEÑO DEL PLAN DE PRUEBAS

1.1. PRUEBAS DE INTEGRACIÓN

Este apartado tiene como objetivo la definición de las pruebas que verificarán si el sistema satisface los requisitos expuestos por el cliente, y que se realizarán antes de la puesta en explotación del mismo, en base a unos criterios de aceptación del sistema que serán definidos prestando atención a: procesos críticos, rendimiento, seguridad y disponibilidad del sistema, atendiendo además a los criterios de calidad marcados en el Plan de gestión de calidad.

El esquema de nombrado para los requisitos de pruebas es el siguiente:

- Un identificador: P-XXXX, donde XXXX representa el número de requisito.
- Una descripción breve del requisito con no más de 3 ó 4 palabras.
- Una explicación más detallada.
- Nivel de aceptación, medida para considerar superada la prueba.
- Los requisitos relacionados con la prueba.

A continuación se presentan las pruebas diseñadas:

Identificador: P-0001	
DESCRIPCIÓN	Comprobar, con otros sniffers, que la captura es la misma
EXPLICACIÓN DETALLADA	1. Arranque del sniffer utilizado en la aplicación y dos más distintos, grabando los 3 el tráfico capturado. 2. Generación de tráfico mediante "*wget* www.terra.es". 3. Detener cualquier generación de tráfico. 4. Comparar las 3 capturas.
HERRAMIENTAS	1. Otros sniffers. 2. "wget www.terra.es" 4. "diff"
NIVEL DE ACEPTACIÓN	Ninguna diferencia a nivel binario entre dos capturas.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0002	
DESCRIPCIÓN	Comprobar que captura todos los paquetes sobre todo cuando pasa de una hora a otra, que no se pierda ninguno.
EXPLICACIÓN DETALLADA	1. Cambiar el reloj del sistema a la misma hora actual y 59 minutos. 2. Arrancar el sistema grabando todo el tráfico 3. Arrancar una instancia de otro sniffer grabando todo el tráfico. 4. Comprobar que aun no se ha llegado a la siguiente hora. 5. Generar tráfico ejecutando u otras páginas web así como otros protocolos hasta pasados 3 minutos por lo menos. 6. Detener cualquier generación de tráfico. 7. Comparar las 3 capturas de tráfico. 8. Restablecer la hora real.
HERRAMIENTAS	3. Otros sniffers. 4."wget www.20minutos.es" 5. "diff"
NIVEL DE ACEPTACIÓN	Ninguna diferencia a nivel binario entre dos capturas.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0003	
DESCRIPCIÓN	Reproducir capturas de ejemplo y ver que llegan iguales.
EXPLICACIÓN DETALLADA	1. Descargar capturas de muestra de los protocolos soportados por la aplicación. 2. Detener cualquier generación de tráfico. 3. Arrancar grabando todo el tráfico el sniffer del sistema. 4. Realizar un "replay" de esas capturas. 5. Comparar la captura descargada con la generada.
HERRAMIENTAS	1. http://wiki.wireshark.org/SampleCaptures 4. replay de captura con tcpdump. 5. "diff"
NIVEL DE ACEPTACIÓN	Por cada protocolo, ninguna diferencia a nivel binario entre dos capturas. *Nota: Las fechas pueden variar.*
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0004	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Comparar las capturas resultantes.
HERRAMIENTAS	2. Otros sniffers. 4.- tcpspray. 7. “mergecap” 8. "diff"
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0005	
DESCRIPCIÓN	Decodificar las capturas de muestra comprobando los datos que se obtienen.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. Arrancar el sistema con decodificación activada. 3. Navegar por una página web. 4. Comparar la decodificación con la web visitada.
HERRAMIENTAS	3. wget www.madrid.org
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0006	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada y decodificación. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Comparar las capturas resultantes. 9. Comparar los contenidos decodificados con lo esperado.
HERRAMIENTAS	2. Otros sniffers. 4.- tcpspray. 7. “mergecap” 8. "diff"
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo. El tráfico decodificado debe contener lo mismo que lo accedido, sin perderse ningún elemento.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0007	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado para encontrar los límites.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada, decodificación y búsqueda de palabras sospechosas. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Comparar las capturas resultantes. 9. Comparar los contenidos decodificados con lo esperado. 10. Comprobar en el log y en la base de datos que las alertas han sido generadas.
HERRAMIENTAS	2. Otros sniffers. 4.- "tcpspray". 7. “mergecap” 8. "diff"
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo. El tráfico decodificado debe contener lo mismo que lo accedido, sin perderse ningún elemento. Las alertas han sido generadas tanto en los logs como en la base de datos.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0008	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado para encontrar los límites, conexión con el operador con Ethernet.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. Arrancar la máquina de Operador. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada, decodificación y búsqueda de palabras sospechosas. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Esperar 15 minutos para que finalice la transferencia entre el PCSniffer y el PCOperador 8. Comparar la captura de la máquina de test con la del PCSniffer y la del PCOperador. 9. Comparar los contenidos decodificados con lo esperado. 10. Comprobar en el log y en la base de datos que las alertas han sido generadas.
HERRAMIENTAS	
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo en las 3 capturas. El tráfico decodificado debe contener lo mismo que lo accedido, sin perderse ningún elemento. Las alertas han sido generadas tanto en los logs como en la base de datos.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0009	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado para encontrar los límites, conexión con el operador con Wifi.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. Arrancar la máquina de Operador. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada, decodificación y búsqueda de palabras sospechosas. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico wifi posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Esperar 15 minutos para que finalice la transferencia entre el PCSniffer y el PCOperador 8. Comparar la captura de la máquina de test con la del PCSniffer y la del PCOperador. 9. Comparar los contenidos decodificados con lo esperado. 10. Comprobar en el log y en la base de datos que las alertas han sido generadas.
HERRAMIENTAS	
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo en las 3 capturas. El tráfico decodificado debe contener lo mismo que lo accedido, sin perderse ningún elemento. Las alertas han sido generadas tanto en los logs como en la base de datos.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0010	
DESCRIPCIÓN	QoS: intentar saturar esta fase con mucho tráfico simulado para encontrar los límites, conexión con el operador con 3G.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. Arrancar la máquina de Operador. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada, decodificación y búsqueda de palabras sospechosas. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. 5. Detener la generación de tráfico en las 2 máquinas. 6. Detener los 3 sniffers. 7. Juntas las capturas de las máquinas de test en una. 8. Esperar 15 minutos para que finalice la transferencia entre el PCSniffer y el PCOperador 8. Comparar la captura de la máquina de test con la del PCSniffer y la del PCOperador. 9. Comparar los contenidos decodificados con lo esperado. 10. Comprobar en el log y en la base de datos que las alertas han sido generadas.
HERRAMIENTAS	
NIVEL DE ACEPTACIÓN	Salvando los cambios de fecha/hora local de cada máquina, el tráfico debe ser el mismo en las 3 capturas. El tráfico decodificado debe contener lo mismo que lo accedido, sin perderse ningún elemento. Las alertas han sido generadas tanto en los logs como en la base de datos.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0011	
DESCRIPCIÓN	Generar varias alertas cuando está enviando un fichero pesado.
EXPLICACIÓN DETALLADA	1. Comprobar que no hay tráfico de red. 2. Arrancar la máquina de Operador. 2. En 2 máquinas distintas pero en la misma red, arrancar un sniffer que grabe sólo el tráfico dirigido o emitido por su propia IP. 3. En la máquina del sistema a testear, arrancar el sniffer con grabación de tráfico activada, decodificación y búsqueda de palabras sospechosas. 4. En las 2 máquinas distintas, generar el máximo volumen de tráfico posible durante 10 minutos. Con palabras sospechosas 5. Comprobar en el log y en la base de datos que las alertas han sido generadas. 6. Enviar un fichero pesado por el mismo canal. 7. Comprobar en el log y en la base de datos que las alertas siguen siendo generadas.
HERRAMIENTAS	
NIVEL DE ACEPTACIÓN	Las alertas generadas deben verse al mismo ritmo en el log del operador que si no se enviase el fichero pesado.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0012	
DESCRIPCIÓN	Dar de alta una palabra mal formada (no saneada).
EXPLICACIÓN DETALLADA	1. Acceder al interfaz web, sección CASO, opción "Agregar" 2. Agregar la siguiente palabra "?%d%s__.()"
HERRAMIENTAS	1. Navegador web
NIVEL DE ACEPTACIÓN	Se obtendrá un mensaje de error y la palabra no habrá sido agregada.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0013	
DESCRIPCIÓN	Parar y arrancar el sistema rápidamente varias veces.
EXPLICACIÓN DETALLADA	1. Acceder al interfaz web, y arrancar el sistema. 2. Inmediatamente, acceder al interfaz web, y parar. 3. Inmediatamente, acceder al interfaz web, y arrancar el sistema. 4. Inmediatamente, acceder al interfaz web, y parar. 5. Inmediatamente, acceder al interfaz web, y arrancar el sistema. 6. Inmediatamente, acceder al interfaz web, y parar.
HERRAMIENTAS	1. Navegador web
NIVEL DE ACEPTACIÓN	Los procesos se arrancan y desparecen consecuentemente con las órdenes dadas.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0014	
DESCRIPCIÓN	Arrancar el sistema con el sistema de ficheros al 98% de ocupación.
EXPLICACIÓN DETALLADA	1. Llenar el sistema de ficheros al 98% 2. Acceder al interfaz web, y arrancar el sistema Snipol.
HERRAMIENTAS	1. Crear uno o varios ficheros grandes con el comando "yes > ficheroGrande".
NIVEL DE ACEPTACIÓN	Se obtendrá una alerta de mantenimiento sobre el estado del disco. El software empezará a eliminar ficheros de capturas, decodificaciones y logs hasta que disponga de suficiente espacio libre (definido en el fichero de configuración) o no tenga más ficheros que borrar.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

Identificador: P-0015	
DESCRIPCIÓN	Testeo completo del sistema y aislado en laboratorio.
EXPLICACIÓN DETALLADA	1. Elegir un técnico independiente a este proyecto. 2. Facilitarle este documento y el software relacionado para que monte la aplicación según las instrucciones de este documento. (apartado 6.2) 3. Ejecutar todos los casos de uso.
HERRAMIENTAS	Las indicadas en el documento de instalación del software.
NIVEL DE ACEPTACIÓN	Todos los casos de uso deben cumplirse satisfactoriamente.
REQUISITOS RELACIONADOS	*No se especifican requisitos relacionados a expensas de validación por el comité de Indect en Marzo de 2010.*

1.2 MATRIZ DE TRAZABILIDAD

Correspondencia entre los requisitos planteados y las pruebas que los cubren.

Pendiente de recibir confirmaciones del comité Indect.

REQ/PRUEBA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															

1.3 PLAN DE PRUEBAS DE ACEPTACIÓN

Previa instalación del sistema por un técnico independiente del proyecto, se realizarán las siguientes pruebas de aceptación.

Identificador: PA-0001	
DESCRIPCIÓN	Descubrimiento de una contraseña en un envío de correo por SMTP
EXPLICACIÓN DETALLADA	se definirá la palabra “pass”, utilizada en el protocolo de negociación de SMTP, en base de datos de palabras sospechosas y se obtendrá en una alerta el aviso de detección de esta palabra, y por el contexto presentado será visible la contraseña.
NIVEL DE ACEPTACIÓN	Descubrir la contraseña de una cuenta de correo al enviar un email por SMTP.
REQUISITOS RELACIONADOS	

Identificador: PA-0002	
DESCRIPCIÓN	Generación de una alerta HTTP
EXPLICACIÓN DETALLADA	Navegar por Internet buscando información de cómo construir una bomba generando una alerta.
NIVEL DE ACEPTACIÓN	La alerta debe llegar al operador antes de 60 segundos.
REQUISITOS RELACIONADOS	

Identificador: PA-0003	
DESCRIPCIÓN	Provocar la generación de una alerta por descargarse un fichero por FTP
EXPLICACIÓN DETALLADA	Provocar la generación de una alerta por descargarse un fichero por FTP llamado “alijo.20091212.zip” y examinar el contenido de ese fichero desde el PC del Operador.
NIVEL DE ACEPTACIÓN	- Recibir la alerta en menos de 10 minutos. - Examinar el contenido de ese fichero desde el PC_Operador.
REQUISITOS RELACIONADOS	

Identificador: PA-0004	
DESCRIPCIÓN	Supervisión completa de correos IMAP.
EXPLICACIÓN DETALLADA	Leer los correos recibidos por IMAP en la última hora.
NIVEL DE ACEPTACIÓN	Lectura de todos los correos recibidos en esa sesión IMAP.
REQUISITOS RELACIONADOS	

Identificador: PA-0005	
DESCRIPCIÓN	Generación de alerta DNS
EXPLICACIÓN DETALLADA	Obtener una alerta impresa porque el usuario intente realizar una conexión de cualquier tipo con el dominio con whitesupremacy.org
NIVEL DE ACEPTACIÓN	Generación de alerta en el log del PC_Sniffer y PC_Operador.
REQUISITOS RELACIONADOS	

Identificador: PA-0006	
DESCRIPCIÓN	Supervisión de correo POP3
EXPLICACIÓN DETALLADA	Conocer el contexto de un mensaje que el usuario ha recibido con la palabra “zulo”.
NIVEL DE ACEPTACIÓN	Generación de alerta en menos de 1 minuto.
REQUISITOS RELACIONADOS	

Identificador: PA-0007	
DESCRIPCIÓN	Supervisión de SIP
EXPLICACIÓN DETALLADA	Escuchar la última llamada realizada por SIP.
NIVEL DE ACEPTACIÓN	Escuchar la última llamada realizada por SIP.
REQUISITOS RELACIONADOS	

1.4 RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN

A continuación se informa del resultado de las pruebas realizadas.

Pruebas a realizar por el comité de Indect.

RESULTADOS DE PRUEBAS			
ID_Prueba	**Título**	**Resultado**	**Notas**
P0001			
P0002			
P0003			
P0004			
P0005			
P0006			
P0007			

Anexo VI: “Tecnologías elegidas para cada componente”.

A continuación se exponen la tecnología elegida finalmente para cada componente a fecha de 5/01/2010.

[c001] Interceptor_DSL:

El diseño de este componente queda excluido de este documento.

[coo1.001] Acoplado a la línea

[coo1.002] Generador de señal

[c002] Router:

El diseño de este componente queda excluido de este documento.

C002.001 Hardware

C002.001.001 Router: N/A en este documento.

C002.002 Software

C002.002.001 Firmware: N/A en este documento.

[c003] PCS ó PCSniffer

C003.001 HARDWARE

[c003.001.001] Microprocesador

Por la proyección que tiene, características técnicas, disponibilidad de software y entornos gráficos compatibles con él y el ser el relevo natural de la arquitectura x86, se elige el procesador **AMD64** para el desarrollo de este proyecto.

C003.001.002: Placa

Siendo compatible con el resto del hardware elegido, se opta por el modelo AMD DB800 , ampliamente utilizado y distribuido, con actualizaciones de firmware en caso de detectarse cualquier problema y compatible también con Coreboot.

C003.001.003: Almacenamiento

Para el almacenamiento de datos se elige el componente LaCie 4big Quadra 8 TB, esperándose en próximas revisiones de este documento el avance de la tecnología “Solid State” para utilizarlo en este proyecto.

C003.001.004, Puerto de conexión auxiliar

Según las comparativas hechas, la relación *velocidad/estándar/abastecimiento* llevan a la elección del estándar **USB-A** para este componente.

C003.001.005, Interfaz de red de entrada

Según las comparativas hechas, la relación v*elocidad/estándar/abastecimiento* llevan a la elección del estándar RJ45 para Ethernet 802.3 en este componente, con anchos de banda de 10/100/1000 MB por seg. Se utilizará la tarjeta Realtek RTL8168C .

C003.001.006, Interfaz de red para conexión con operador.

Según la decisión del operador evaluando sobre el terreno la instalación a realizar, se utilizará una interfaz Ethernet, Wifi54G ó 3G.

C003.001.007, RAM

Se utilizará la siguiente combinación de memorias:

DDR3 24.576 MB 1600Mhz	2 DIMMS	47.152 MB

Estimándose esa cantidad de memoria RAM como útil para filtrar un DSLam. No obstante, en función de presupuestos, esta cantidad es fácilmente aumentable a 98 GB.

C003.001.8 Sistema de alimentación eléctrica

Se utilizará un SAI “Powerware 5125 1500VA” para proporcionar hasta 47h. de alimentación eléctrica ininterrumpida.

[c003.002] SOFTWARE

C003.002.001, Sistema operativo:

Se elige **Debian 5.0** por su estabilidad, seguridad, fiabilidad y gran proceso de depuración de su software que a menudo dura meses e incluso años antes de pasarlo a producción, así como el amplio tiempo de mantenimiento de cada versión.

C003.002.002: Lenguaje de programación para el binario..

Según las estadísticas de rendimiento obtenidas, se decide utilizar **C++** como lenguaje de programación para aumentar el rendimiento.

C003.002.003: Sistema de soporte remoto.

Por ser el protocolo de acceso remoto cifrado más conocido, extendido y testado, se escoge SSH para este componente.

C003.002.004: Sistema de ficheros y cifrado.

Se utilizará el sistema de ficheros ext4 con LUKS como módulo de cifrado.

C003.002.005, Sniffer

Se utilizará tcpdump 4.0 (Octubre 27, 2008).

C003.002.006, Decodificador.

Se utilizará Xplico por ser una solución que cubre las necesidades obligatorias.

C003.002.007, Analizador de datos.

- **Análisis de palabras**: se utilizará la técnica de **Bloom Filters** por ser considerablemente más rápida que la ejecución de continuas peticiones SQL.

- **Librería para generación de hash**: Se opta por **SHA1** al ser un método más moderno que minimiza bastante más, frente a **MD5**, una colisión del hash de dos palabras. No obstante, teniendo en cuenta el consumo de CPU mayor frente a md5, se valorará su reemplazo por éste si se producen problemas de rendimiento.

- **API para la conexión con la BD (Sqlite3)**: Se utilizará la versión oficial de Sqlite3 para conectar con la base de datos desde C++.

C003.002.008, compresor

La decisión de este componente es importante, ya que se utilizará gran parte de la CPU en esta función. Las dos variables en juego son el ancho de banda para enviar las capturas (upload) y la CPU disponible. Por tanto el objetivo es enviar las capturas lo más cercano al tiempo real, contando que el canal de subida pueda ser reducido, pero sin saturar el procesador, lo que podría suponer la pérdida ó demora de detección de alertas.

Se elige **Pigz** pues, aunque pbzip2 tiene un mejor ratio de compresión, el tiempo empleado es aproximadamente un 50% mayor, lo cual no compensa para una diferencia de apenas un 4% frente a Pigz.

C003.002.009, software de envío de datos (SCP)

Se utilizará SCP como software de envío de datos.

C003.002.010: Base de datos.

Debido a las condiciones tanto técnicas como legales de las bases de datos estudiadas, se opta por **Sqlite** v3 como SGBD para este proyecto.

C003.002.011: Sistema de impresión

En la mayoría de las distribuciones GNU/Linux el sistema de impresión instalado es **CUPS** (Common Unix Printing System). Es el más potente actual así como el que mejor soporte y mantenimiento lleva, por lo que es la opción preferente para este proyecto.

C002.002.012: Lenguaje de programación para el interfaz.

Se utilizará **PHP** como lenguaje sencillo y confiable para realizar interfaces web. Así mismo se hará apoyo en el framework Cakephp para despliegue rápido de aplicaciones web.

[c004] ServidorDeAlmacenamiento:

C004.001 HARDWARE

c004.001.001 Microprocesador:

Cualquier procesador superior a AMD Phenom 9500 QuadCore 2.2 Ghz es válido.

C004.001.002 Placa base:

Cualquier placa base compatible con los componentes [c004.001.001] Microprocesador y C004.001.003, Almacenamiento es válida. Deseable compatible con Coreboot.

C004.001.003, Almacenamiento:

Pese a que la capacidad mínima exigida son 5.035'8 GB, se utilizará el disco duro externo en modo NAS LaCie 5big Network 10 TB.

C004.001.004, Puerto de conexión auxiliar

Se utilizará el mismo tipo de puerto de conexión auxiliar que en el PCSniffer, el formato USB.

C004.001.005, Interfaz de red de entrada:

Para recibir los datos que envíe el PCSniffer se utilizará el mismo tipo de interfaz por el cual son emitidos en el primer equipo, quedando por tanto a elección del Operador entre Ethernet, Wifi54G ó 3G.

C004.001.006, Interfaz de red para conexión con Internet/Intranet

A elección del operador.

C004.001.007, teclado y ratón

A elección del operador.

[C004.002 SOFTWARE]

C004.002.001, Sistema operativo:

Se empleará el sistema operativo **Debian 5.0.** .

C004.002.001: Sistema de impresión.

Igual que en el componente C003.002.011, se utilizará **CUPS**.

C004.002.003: Sistema de ficheros. Ext4

Se usará **ext4** y cifrado mediante **LUKS**.

C004.002.004: Sistema de interfaz gráfica de usuario

Se utilizará indistintamente una interfaz gráfica como Gnome o KDE. Se preferirá **KDE** por ser más fácil para los usuarios provenientes de Ms. Windows. En la versión actual del proyecto se instalará la versión 3.x, al no estar disponible la rama 4.x en los repositorios oficiales, la cual habría sido preferible al facilitar más el mantenimiento del proyecto a largo plazo.

C004.002.005: Navegador web.

Se utilizará **Mozilla Firefox 3.5.x** para acceder al interfaz web de la aplicación.

C005.001.001: Impresora

Se utilizará una impresora pequeña como la mencionada **HP OfficeJet H470**.