

12. Conclusiones y líneas futuras de trabajo

Esta última sección de la memoria de proyecto está dedicada a exponer las conclusiones finales de nuestro trabajo. Enunciamos a continuación las aportaciones (las que consideramos más importantes), de este proyecto al estado del arte actual. Después algunas de las aportaciones personales obtenidas durante la ejecución del proyecto. Y finalmente exponemos las líneas futuras de trabajo posibles.

Consideramos que el presente Proyecto aporta al “*Estado del Arte*” actual:

- Un sistema que integra módulos de distinta naturaleza y función, consiguiendo establecer una estrecha relación entre ellos.
Ello es fruto del sistema de comunicación entre módulos implementado, que permite que entre ellos compatibilicen ciertas funcionalidades mediante la compartición de datos de configuración.
- El empleo de la potencia que ofrece FireHOL, en un sistema de estas características, algo que hasta ahora no se había realizado.
Ello le confiere cierto grado de innovación. Sólo un uso más práctico del sistema nos dirá si la elección ha sido la acertada.
- La permisividad/flexibilidad del sistema al adaptarse a los cambios que manualmente pueda realizar el usuario a los ficheros de configuración (se entiende que mediante un editor de texto, u otras herramientas externas).
Se ha tenido especial cuidado en que la interfaz de configuración respete el formato original de los ficheros de configuración, y sea capaz de adaptarse a posibles modificaciones realizadas en los mismos. Ello permite al usuario realizar (hasta cierto punto) cuantos cambios desee en dichos ficheros, que serán tenidos en cuenta y preservados por el sistema de configuración.
- La ventaja de emplear un sistema operativo totalmente estándar (Debian ‘*Sarge*’ 3.1) y actual, siendo capaz de soportar actualizaciones o adición de nuevos paquetes según el usuario lo necesite.⁴
- Al estar el código extensivamente documentado, y abrazar los principios de la modularidad y extensibilidad en su construcción, se hace factible modificarlo posteriormente sin excesiva dificultad (aunque el desarrollador no llegase a leerse ni una línea de ésta documentación, cosa más que probable...)

La realización de este proyecto nos ha permitido desarrollar habilidades y ampliar nuestros conocimientos en diversos campos. A continuación resumimos algunas de las aportaciones personales con las que el desarrollo de este proyecto nos ha enriquecido:

- Obtención de conceptos generales de seguridad de la información.
- Conocimiento de herramientas (software) de seguridad perimetral.
- Adquisición de experiencia en la integración de aplicaciones en entornos Unix/Linux.
- Adquisición de experiencia en el desarrollo de aplicaciones Web.
- Profundización en el conocimiento del lenguaje de programación PHP.

⁴Siempre que no rompa las dependencias existentes, ni se instalen nuevas versiones de los paquetes que se han modificado expresamente para este sistema. Así mismo, es conveniente conservar los ficheros de configuración de los módulos tal y como han sido suministrados.

- Programación orientada a objetos con el lenguaje PHP.
- Profundización en el conocimiento del lenguaje extensible de etiquetas XML y su procesado automático.
- Incremento de conocimiento sobre la estructura interna de los sistemas operativos Unix/Linux.
- Generación de paquetes Debian.
- Creación de distribuciones personalizadas a partir de sistemas Debian.

Líneas Futuras de Trabajo

El Proyecto admite gran cantidad de adiciones y continuaciones dado que su concepción es, a priori, muy amplia. Su arquitectura modular admite la adición de nuevas aplicaciones. La cantidad de este tipo de herramientas de seguridad o monitorización que hay disponibles, permiten que se tenga un número *casi* infinito de posibilidades a la hora de elegir más incorporaciones. También son factibles la realización de mejoras que automaticen aún más la gestión de la configuración de dichas aplicaciones, y su relación con el resto del sistema.

A continuación presentamos algunas posibilidades de mejora para el sistema actual:

1. Incorporar una interfaz de configuración global de la aplicación: para permitir configurar parámetros globales de la aplicación, tales como las rutas de los programas empleados, el idioma/localización por defecto, la elección de módulos a emplear (en caso de que haya varias alternativas), etc...
2. Separar por completo la «presentación» del código PHP, ya que en ocasiones se generan pequeñas cadenas de código HTML dentro de éste. Así mismo, independizar el código hasta donde sea posible de las interioridades del protocolo HTTP, como por ejemplo, allá donde se realiza gestión directa de las variables GET o POST.
3. Mejorar el manejo de situaciones, que actualmente se limitan a generar un mensaje de error, cuando lo deseable sería que el sistema se “autorecuperara” en la medida de lo posible de las mismas, como por ejemplo: Ficheros de configuración inexistentes que podrían ser recreados desde cero, parámetros imprescindibles que el usuario deba introducir por primera vez, etc.
4. Permitir la gestión, por parte de patForms, de más de un botón de *Submit* por formulario.⁵
5. Mejoras a los módulos:
 - a) Antivirus: pasar al Proxy sus parámetros de “autoconfiguración”, de forma que este último no tenga que saber cómo interactuar expresamente con este tipo de Antivirus para poder incorporarlo a su configuración.
 - b) Filtro de Contenidos (DansGuardian): permitir añadir nuevos grupos de Usuarios, o de Filtrado.
 - c) Cortafuegos (FireHOL): independizar completamente del nombre de las interfaces de red del sistema.⁶
 - d) IDS (Snort): optimizar aún más la lectura de los ficheros de reglas. Permitir una edición gráfica de las reglas, y/o comprobar su sintaxis.
6. Mejorar a la “Metadistribución”. Añadir un instalador gráfico, ampliar el hardware soportado y actualizar periódicamente los paquetes instalados en ella.

⁵patForms en estos momentos no lo soporta. A nivel de código de la aplicación se ha intentado implementar esta posibilidad, sin llegar a terminarlo.

⁶Actualmente se ha asumido que las interfaces son de la forma `eth*`