Detección de posibles anomalías en la Infraestructura de Clave Pública (PKI) tipo RSA por medio de análisis probabilísticos y estadísticos.

Ing. Antonio Castro Lechtaler[1,2] Msc.
Esp. Marcelo Cipriano[1], Lic. Edith García[1],Lic. Julio Liporace[1]
Lic. Ariel Maiorano[1], Eduardo Malvacio[1]

[1]Laboratorio de Investigación en Técnicas Criptográficas y Seguridad Teleinformática.
Escuela Superior Técnica, Facultad de Ingeniería. Instituto Universitario del Ejército.
[2] F.C.E. Universidad de Buenos Aires.

acastro@iese.edu.ar
marcelocipriano@iese.edu.ar
edumalvacio@gmail.com

1. Resumen.

Esta línea de investigación persigue la elaboración de herramientas matemáticas susceptibles de ser sistematizadas en un software capaz de detectar anomalías y mal funcionamiento en servicios de infraestructura de clave pública (PKI) que utilicen el esquema RSA.

Esta herramienta se puede aplicar en redes Públicas o Privados, Lan´s, o Wan´s e incluso Internet; en sistemas militares como del ámbito civil.

Errores – fortuitos o involucrados [1] - o mal funcionamiento en el software que provee de autenticación y confidencialidad provocan debilitamiento de los sistemas que la PKI intenta proteger.

El esquema RSA[1] utiliza módulos públicos producto de 2 números primos grandes – 512, 1024, 2048 e inclusive 4096 bits de longitud o mayores aún- intratable de factorizar con los actuales métodos. Sin embargo, un sesgo en la selección de estos valores podría provocar una brecha en la seguridad.

Esta herramienta informática que estamos elaborando permitiría la detección de mal funcionamiento en la distribución probabilística de los factores primos. Sin embargo dada la astronómica cantidad de números posibles, la detección es estadística, mediante el estudio de muestras suministradas por la PKI en observación.

2. Palabras Claves.

Seguridad en Redes, Infraestructura de Clave Pública, PKI, Detección de Anomalías, Open-SSL, RSA.

3. Contexto.

El Laboratorio de Investigación en Técnicas Criptográficas y Seguridad Teleinformática (CRIPTOLAB) pertenece a la Escuela Superior Técnica “Gral. Div. Manuel N. Savio” (EST), Facultad

[1] La 3-tupla (*n*,*e*,*d*): n (módulo) es el producto de 2 primos, e (clave pública) y d (clave privada) son inversos entre sí mód ϕ(n).

de Ingeniería, del Instituto Universitario del Ejército Argentino (IUE) en el área del Posgrado en Criptografía y Seguridad Informática que se dicta en esta institución, junto a otros posgrados y carreras de grado en ingeniería.

El desarrollo científico y tecnológico es relevante a nivel estratégico y es por ello que tanto las Fuerzas Armadas en general como el Ejército en particular.

El Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF) realizó aportes a través de Proyectos de Investigación Científico Tecnológicos Orientados (PICTO) para la realización durante 6 años del proyecto recientemente finalizado sobre "REDES PRIVADAS COMUNITARIAS".

Dentro de dicho proyecto, se llevó adelante gran parte de esta investigación-, cuyos resultados parciales o finales han sido presentados en varios CACIC para su difusión a la comunidad científica.

Sin embargo hemos podido, recién-temente, darle entidad propia a esta línea de investigación al poder incluirla dentro de los proyectos de la EST – IUE bajo el nombre de VULCLAP: Vulnerabilidades en Clave Pública.

Asimismo tiene posibilidades de ser inscripta en la reciente ampliación de la convocatoria del Ministerio de Ciencia, Técnica e Innovación Productiva, para proyectos de investigación, como consecuencia de la creación del Instituto Universitario del Ejército.

4. Introducción.

4.1. Generalidades

La Criptografía dejó de pertenecer a la esfera de los secretos militares y diplomáticos y se ha volcado al ámbito civil en general. No han dejado de crecer sus aplicaciones: ingreso a redes sociales, servidores de mails, home banking, compras online, redes WI-FI, almacenamiento de información confidencial o sensible, etc.

Todo lo que implique intercambio de información entre dos equipos informáticos, debería estar al resguardo de ojos indiscretos, sobre todo los intercambios inalámbricos.

Los sistemas se encargan de entablar todos los servicios requeridos por los estándares y protocolos de manera automática: tickets de ingreso a sistemas, intercambios de claves, autenticación de usuarios y equipos, inicio de sesión, y muchos otros son controlados en forma automática. Los usuarios por lo general no cambian las opciones por defecto: por ejemplo las personas que instalan sus routers con servicios inalámbricos sin autenticación de usuarios ni protocolos de encripción.

¿Cómo comprobar si estos procesos y servicios contienen errores que pueden alterar la seguridad de lo que pretenden proteger, sin leer el código fuente de los mismos?

4.2. Código abierto no necesariamente aumenta la Seguridad de los Sistemas de Información.

La filosofía del código abierto (open source) y la ley de Linux [2] es atractiva

desde cierto punto de vista teórico. Pero por sí misma no garantiza la ausencia de errores. Una muestra de ello es el bug descubierto por Luciano Bello en OpenSSL[2] de Debian. El mismo fue enmendado 20 meses después que la versión defectuosa fuera informada [3].

4.3. Planteamiento del problema.

De acuerdo a lo expuesto en 4.2. proponemos la existencia de una mecánica adicional de control.

Una PKI tipo Open-SSL que ofreciera protección criptográfica (en sus diferentes formas antes mencionadas: claves de sesión, autenticación de equipos, etc.) es vulnerable cuando:

1) genera un número relativamente pequeño de primos[3].

2) está diseñada con una directiva errónea para la selección de los primos o la generación de los módulos.

En ambos casos la PKI se distancia del comportamiento equiprobable, dentro de ciertos parámetros asumidos, a un comportamiento sesgado. Susceptible de ser estudiado por un atacante y vulnerar así el sistema.

Así, un atacante tiene forma de construir un subconjunto P' de números primos con los que trabaja tal que el conocimiento de este conjunto le permita vulnerar la factorización de RSA para una cantidad significativa de módulos.

[2] Una mala inicialización de una variable provocó una predictibilidad en el generador de números, abriendo una vulnerabilidad inimaginable.

[3] Vulnerabilidad de OpenSSL de Devian descubierta por L. Bello.

Vemos que se deben hallar los factores primos del esquema RSA, proceso complejo pues esa es la fortaleza que permite a esta criptosistema proteger la información: la complejidad en la factorización de un número enorme.

Sabemos de Lenstra [4] y otros investigadores de reputación reconocida, aunque partieron de otras hipótesis y escenarios de investigación, han hallado colisiones de primos en el 5% de una gran muestra de módulos públicos RSA de 1024 bits y por ende vulnerabilidades en los mismos. Ello nos anima a seguir investigando al respecto.

5. Resultados y Objetivos.

5.1. Resultados intermedios de la línea de Investigación.

En el año 2008 nuestra investigación se oriento a la elaboración una herramienta informática que permite hallar los primos que componen un módulo RSA con la información que aportan su clave pública y su clave privada, resultado presentado y publicado en CACIC 2008 [5] realizado en La Rioja, Argentina.

Las posteriores pruebas de codificación e implementación demostraron que este procedimiento corría muy veloz presentándose estos resultados en CUBA [6].

Seguimos aún estudiando el comportamiento de este algoritmo y lo comparamos con el procedimiento que aborda el mismo problema y lo resuelve, existente en la bibliografía tradicional para la enseñanza de la criptografía [7].

Los análisis indicaron que la complejidad computacional de nuestro algoritmo era

del orden O (log n) mientras que el de la bibliografía de referencia tenía un orden O ($\log^3$ n). Resultado presentado en Chile y que nos sorprendió de manera grata [8].

Hallada la herramienta matemática que permitiría detectar anomalías (en caso que las hubiere) el resultado publicado en CACIC 2011 [9] en La Plata.

Con el abordaje probabilístico del problema presentado en 41 JAIIO [10] y por último el diseño final de la herramienta probabilística y estadística presentado en CACIC 2012 [11] en Bahía Blanca se terminó de dar forma definitiva al sustento teórico.

Simultáneamente al avance matemático se ensamblaron y codificaron todas las herramientas matemáticas antes mencionadas, en una plataforma de software programado en C++.

5.2. Etapa actual

Se llevaron adelante pruebas en las que, por lotes, se le solicita a OPEN-SSL la entrega de módulos RSA a efectos de buscar colisiones de primos. Es decir, se analizan los factores primos de cada uno de ellos y se los compara con los de los otros módulos del lote en busca de factores repetidos.

En primer lugar estamos buscando las causas por las cuales el proceso informático agota la memoria de la computadora y el sistema operativo interrumpe la ejecución del programa. Aún no encontramos la causa y es por ello que la exploración sólo se ha realizado en lotes de 1000 módulos cada uno y para 64 bits de tamaño de los mismos.

Pese a esta dificultad se han podido evaluar 40.000.000 de módulos agrupados en 40000 lotes de 1000 cada uno. Los resultados obtenidos siguen *sin responder a las predicciones teóricas*, lo cual no nos permitiría aún aseverar que la PKI de estudio está sesgada o que todos los segmentos del modelo matemático – informático están funcionando bien. Estas dificultades han hecho que se demore el tiempo planteado inicialmente para estas etapas dentro de la línea de investigación.

Actualmente estamos revisando todo en búsqueda de respuestas a las inconsistencias halladas. Hay 3 posibilidades, las cuales están siendo revisadas en este momento:

a) el modelo matemático elegido para la detección de anomalías en la distribución de los primos no es el adecuado y debiera tener modificaciones.

b) el software que busca la detección de una Anomalía en la distribución de factores primos tiene algún error.

c) La versión de OpenSSL que estamos usando como PKI no está programado para otorgar tan velozmente los certificados que pedimos y en la cantidad que le exigimos al hacer los test.

5.3. Etapa final.

Una vez que aseguremos la ausencia de errores, en caso que los hubiera (en cualquiera de las etapas), desarrollaremos un software de auditoría para que detecte anomalías.

Para probar dicho software simularemos diferentes PKI´s con vulnerabilidades y sin ellas. Cada una de ellas en un orden aleatorio, será testeada por el programa. El comportamiento esperado es que

detecte las vulnerables e informe al respecto.

También haremos estudios de "falsos positivos y negativos".

No debemos perder de vista que la herramienta que se intenta desarrollar no trabaja con el universo completo de primos disponibles. Sino con una muestra o conjunto de muestras, que permita inferir una probabilidad de vulnerabilidad. La herramienta matemática que sustenta al software es probabilístico-estadística.

6. Formación de Recursos Humanos.

En el año 2012 algunos algoritmos que utilizamos en esta investigación fueron codificados y probados en el contexto de la Cátedra de Computación I a cargo del Ing. Mg. Alejandro Repetto, que posee nuestra facultad en la carrera de Ingeniería Informática.

En el año 2013 se hicieron las pruebas y pronto podrán implementarse en una plataforma de computación distribuida a los efectos de acelerar la investigación, dada la posibilidad de paralelizar la solución al problema planteado.

Asimismo Eduardo Malvacio, alumno de último año de dicha carrera, se sumó al equipo de investigadores y nos brinda apoyatura en la programación e implementación del software.

7. Referencias y Bibliografía

[01] Young A and Yung M. An Elliptic Curve Asymmetric Backdoor in Open-SSL RSA Key Generation. Chapter 10. Cryptovirology. 2006. http://www.cryptovirology.com.

[02] Glass, Robert "*Facts and Fallacies of Software Engineering"*. Addison-Wesley Professional, 2003.

[03] Bello L, Bertacchini M. "*Generador de Números Pseudo-Aleatorios Predecible en Debian"*. III Encuentro Internacional de Seguridad Informática. Manizales, Colombia. Octubre 2009.

[04] Lenstra, A; Hughes, J; Augier, M y otros. *Ron was wrong, Whit is right*. e-print International Association for Cryptologic Research. 15 Feb 2012 http://eprint.iacr.org/2012/064,

[05] Cipriano, M. "*Factorización de N: recuperación de factores primos a partir de las claves pública y privada*." Anales del XIV Congreso Argentino de Ciencias de la Computación CACIC 2008. Chilecito, La Rioja, Octubre 2008.

[06] Castro Lechtaler, C; Cipriano, M; Benaben A; Quiroga, P. "*Study on the effectiveness and efficiency of an algorithm to factorize N given e and d"* Anales del IX Seminario Iberoamericano en Seguridad de las Tecnologías de la Información, La Habana, CUBA. 2009.

[07] Menezes, A; van Oorschot, P and Vanstone, S. *Handbook of Applied Cryptography*. CRC Press. 5th Edition, 2001.

[08] Benaben, A; Castro Lechtaler, A; Cipriano, M; Foti, A. "*Development, testing and performance evaluation of factoring algorithms whit additional information*" XXVIII Conferencia Internacional de la Sociedad Chilena de Computación. Santiago de Chile. 2009.

[09] Castro Lechtaler, A; Cipriano, M. "*Detección de anomalías en Oráculos tipo OpenSSL por medio del análisis de probabilidades*" Anales del XVII Congreso Argentino de Ciencias de la Computación CACIC 2011. La Plata, Buenos Aires, Octubre 2011.

[10] Castro Lechtaler, Antonio, Cipriano Marcelo; Malvacio Eduardo; Cañón, Sebastián; *Procedure for the Detection of Anomalies in Public Key Infrastruture (RSA Systems).* Anales del XIII Simposio Argentino de Tecnología, 41 Jornadas Argentinas de Informática e Investigación Operativa JAIIO – SADIO. La Plata, Buenos Aires, Agosto 2012.

[11] Castro Lechtaler, Antonio; Cipriano, Marcelo; Malvacio, Eduardo. *Experimental detection of anomalies in public key infrastructure* . Anales del XVIII Congreso Argentino de Ciencias de la Computación CACIC 2012. Bahía Blanca, Buenos Aires, Octubre 2011.