Rekayasa Keamanan pada Kompilator Online untuk Pemrograman Paralel

Andria Arisal
P2Informatika-LIPI
andria.arisal@informatika.lipi.go.id

Taufiq Wirahman
P2Informatika-LIPI
taufiq@informatika.lipi.go.id

Nuryani
P2Informatika-LIPI
nuryani@informatika.lipi.go.id

Wiwin Suwarningsih
P2Informatika-LIPI
wiwin@informatika.lipi.go.id

Abstrak

Kami mengembangkan kompilator online untuk pemrogram parallel sebagai suatu kakas pembelajaran untuk mempraktekkan dan memahami paradigm pemrograman paralel. Kakas ini memiliki antarmuka langsung dengan pengguna umum melalui Internet, sehingga selain mudah digunakan, akan sangat rentan terhadap berbagai macam serangan online. Kami menyadari bahwa factor keamanan adalah salah satu aspek penting dari pengembangan perangkat lunak yang harus diterapkan pada pengembangan kompilator online tersebut. Dengan menganalisis kelamahan dan serangan yang mungkin terjadi, kompilator online ini dibangun dengan menggunakan dua tahapan pengamanan. Pengamanan tahap pertama adalah sebagai bagian dari aplikasi web yang merupakan antarmuka pengamanan terhadap aplikasi web. Pengamanan kedua adalah pengamanan untuk melindungi cluster computer sebagai sumber daya yang digunakan untuk mengeksekusi aplikasi yang dibuat dengan paradigm pemrograman parallel. Kedua lapisan pengamanan ini dianggap cukup dapat mengamankan aplikasi dan sumber daya dari pengguna atau kode yang tidak diharapkan.

Kata kunci: kompilator online, keamanan, kerentanan, serangan, cluster, aplikasi web, pemrograman paralel

1. Pendahuluan

Sewaktu pengembangan suatu aplikasi perangkat lunak, pengembang biasanya mengabaikan aspek kemanan dari perangkat lunak yang dikembangkan. Keamanan dari perangkat lunak adalah salah satu sifat yang menentukan kualitas dari perangkat lunak. Hal ini berhubungan dengan bagaimana perangkat lunak dapat terlindungi dari pengguna atau kode yang berbahaya, yang dapat merusak data internal, fungsi, dan keseluruhan sistem dari pengakat lunak ketika perangkat lunak tersebut diimplementasikan [1,2].

Terdapat tiga kutub yang menentukan aspek kualitas dari rekayasa perangkat lunak; *usability* (kebergunaan), *security* (keamanan), *functionality* (fungsionalitas). *Usability* berhubungan dengan bagaimana pengguna dapat menggunakan perangkat lunak dengan mudah, sedangkan *functionality* adalah bagaimana perangkat lunak yang dikembangkan dapat menjalankan fungsinya dan berprilaku sesuai dengan yang diharapkan. Kedua fungsi tersebut dapat langsung diraskan oleh pengguna ketika pengujian sistem. Sedangkan aspek *security* (keamanan) biasanya diabaikan, karena tidak hanya sulit untuk diimplementasikan tetapi juga dapat mengurangi kemudahan pengguna dalam menggunakan sistem. Keseimbangan antar ketiga aspek tersebut merupakan pertimbangan yang sulit bagi pengembang perangkat lunak.

Kami membangun kompilator paralel online[3] sebagai kakas pendidikan untuk memprakterkkan dan memahami cara

pemrograman dengan menggunakan paradigma pemrograman paralel. Sebagai suatu aplikasi web yang memiliki antarmuka langsung dengan pengguna yang tidak dikenal dan memiliki komunikasi internal dengan lingkungan komputer paralel (cluster), mengakibatkan aplikasi ini menjadi sangat rentan terhadap semua ancaman serangan online. Oleh sebab itu, aspek sekurity harus dipertimbangkan dalam pengembangannya untuk meningkatkan keamanan sistem secara keseluruhan[4].

2. Rekayasa keamanan pada aplikasi online

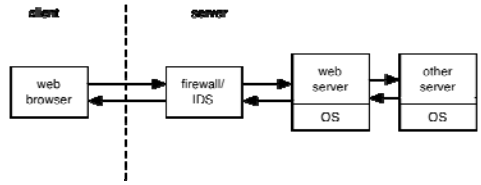
Keamanan sistem meliputi tujuh konsep kunci yang harus dikelola selama sistem tersebut ada. Konsep-konesp tersebut adalah[5]:

- *Authentication*
- *Authorization*
- *Confidentiality*
- *Data/message integrity*
- *Accountability*
- *Availability*
- *Non-repudiation*

Untuk mengelola dan melindungi semua konsep keamanan tersebut, pengembang perangkat lunak seharusnya melakukan aktifitas rekayasa keamanan. Aktifitas tersebut meliputi tujuh aktifias dasar, yaitu[4]:

- Menetapkan tujuan, yaitu menentukan apa yang harus dilindungi, dan lingkungan system apakah yang sudah terlindungi
- Memodelkan ancaman, yaitu mencoba mengenali apa yang dapat terjadi (sebagai akibat dari) jika sesuatu (yang buruk) terjadi
- Arah perancangan, yaitu mendefinisikan apa, mengapa, dan bagaimana melindungi dan mengamankan system dari ancaman yang dimodelkan
- Peninjauan arsitektur dan perancangan, ialah untuk menentukan prioritas dari aspek keamanan yang harus diambil
- Peninjauan kode
- Pengujian
- Peninjauan pengembangan

Aplikasi online adalah sasaran yang populer dari serangan keamanan karena memiliki antarmuka langsung dengan pengguna anonim melalui koneksi internet. Penyerangan keamanan secara online juga semakin bervariasi, mulai dari *sql injection* sampai *cross-site scripting*[6]. Sekenario penyerangan aplikasi secara online dapat digambarkan sebagai gambar 1, dimana pengguna mengirimkan pesan melalui *firewall* atau sistem pendeteksi (*intrussion detection system*) ke server web yang berkomunikasi dengan server lainnya (jika diperlukan) atau langsung menampilkan hasil dan mengirimkan kembali ke pengirim (*web browser*).



Gambar 1 Sekenario penyerangan online yang umum [5][7]

Untuk meningkatkan keamanan aplikasi online, terdapat sembilan kategoi yang harus diperhatikan, antara lain[4]:

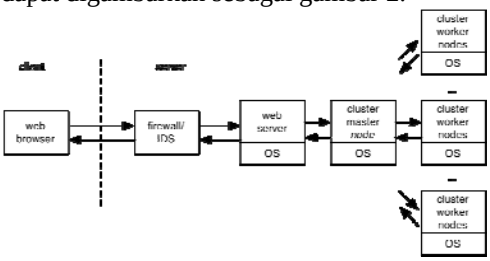
- Validasi data masukan
- *Authentication*
- *Authorization*
- Manajemen konfigurasi
- Kriptografi
- Pengamanan data sensitive
- Manajemen eksepsi
- Manajemen sesi
- *Auditing* dan *logging*

Beberapa dari aspek tersebut yang berhubungan dengan pengamanan kompilator online akan didiskusikan pada bagian berikutnya.

3. Aspek keamanan pada kompilator online untuk pemrograman paralel

Kompilator online untuk pemrograman paralel[3] adalah suatu aplikasi web yang cukup menarik. Aplikasi ini memiliki antarmuka langsungde nan pengguna

melalui web yang juga memiliki komunikasi dengan sistem komputer paralel (cluster) sebagai lingkungan pemrogaraman dan eksekusinya. Diagram sistem secara umum dapat digambarkan sebagai gambar 2.



Gambar 2 Diagram system kompilator online untuk pemrograman paralel

Sistem harus dapat berkumunikasi tidak hanya dengan aplikasi web server, tetapi juga berhubungan pada level sistem operasi melalui *message parsing*. Hal ini mendorong timbulknya banyak masalah dan kelemanan, yang harus diselesaikan untuk mengamankan seluruh sistem (aplikasi dan sistem cluster).

Pengguna berkomunikasi dengan kompilator online memalui antar muka web. Pengguna dapat menuliskan kode program paralel dalam bahasa C/C++ dengan pustaka openmpi pada antar muka web tersebut atau dengan mengunggah berkas berisi kode paralel ke sistem melalui antar muka web. Kode C/C++ dikenal sebagai bahasa pemrograman yang rentan dan berbahaya jika tidak digunakan dengan benar, karena dapat berkomunikasi dengan pustaka sistem operasi pada level yang rendah dan memiliki manajemen memori yang bebas. Hal ini mengakibatkan ancaman bagi seluruh sistem, jika kode program tersebut tidak dikelola dengan baik.

Sehubungan dengan konsep keamanan yang sudah diuraikan pada bagian sebelumnya, kami menyimpulkan bahwa terdapat beberapa ancaman potensial terhadap sistem.

- *Authentication*
  Ancaman memungkinkan pengguna yang tidak otentik dapat mengirimkan, mengkompilasi dan mengeksekusi kode yang dibuatnya melalui antar muka web. Selain itu ancaman tipuan (*spoofing threat*) dapat menipu system untuk mengenali pengguna yang tidak otentik sebagai salah satu pengguna yang asli dari system tersebut.
- *Authorization*
  Ancaman yang memungkinkan pengguna yang tidak sah untuk membuat, mengkompilasi dan menjalankan kode yang jahat dengan menggunakan antar muka web. Ancaman ini juga berhubungan dengan ancaman *spoofing* yang memungkinkan pengguna yang tidak sah tersebut berlakuk sebagai pengguna yang otentik yang diperbolehkan untuk menjalankan aksi yang tidak aman.
- *Confidentiality*
  Karena ditujukan untuk pengajaran, kerahasiaan data pada sistem terhadap kode yang dibuat oleh pengguna dianggap tidak begitu penting.
- *Data/message integrity*
  Integritas data/pesan khususnya antara web server dan sistem cluster sangat diperluukan, terutama pada kode sumber paralel yang dibuat pada antar muka web untuk dijalankan pada sistem cluster. Jika integritas data/pesan dapat dilemahkan, maka terdapat kemungkinan bahwa kode yang dijalankan pada sistem cluster adalah kode yang jahat dan pesan yang dikirimkan adalah untuk maksud yang merusak.
- *Accountability*
  Akuntabilitas sistem diperlukan untuk mengamati kemungkinan aktifitas pengguna yang bermaksud buruk jika berhasil masuk dan menyusupi sistem.
- *Availability*
  Ktersediaan sistem (terutama sistem cluster) akan terancam jika pengguna dapat membuat dan menjalankan program paralel yang menghabiskan sumber daya cluster, sehingga membuat sistem tidak dapat digunakan oleh pengguna lainnya.
- *Non-repudiation*

Karena sistem ini ditujukan untuk penggunaan pengajaran dan berhubungan dengan aktifitas yang *non-repudiation*, sehingga tidak begitu berpengaruh terhadap pengguna lainnya.

4. Pendekatan kami

Selain meningkatkan keamanan sistem, kami harus selalu mempertimbangkan aspek kulalitas lainnya (fungionalitas dan usabilitas). Oleh karena itu, kami harus mengambil perimbangan dalam pengembangan kompilator online untuk perangkat lunak.

Kami menggunakan basis dari aktifitas rekayasa pengamanan perangkat lunak yang diuraikan pada bagian 2 untuk mengamankan aplikasi kompilator online untuk pemrograman paralel yang dikembangkan, yang meliputi pengamanan aplikasi web dan web server sebagai antar muka langsung dengan pengguna dan sistem cluster sebagai lingkungan eksekusi program paralel yang dibuat. Sistem yang dikelola meliputi sistem perangkat keras (komputer, antar muka jaringan, dll), serta perangkat lunak (sistem operasi, aplikasi, pustaka, kode yang disimpan pengguna, dll). Pengguna sistem dapat dikelompokkan menjadi; pengguna tamu (*guest*), pengguna terdaftar (*registered user*), dan administrator. Semua pengguna tamu saling berbagi lingkungan dan ruang direktori yang sama, sehinggan pengguna yang satu dapat melihat, mengkompilasi dan menjalankan kode pengguna tamu lainnya. Hal ini menyebabkan masalah konsistensi dimana dimungkinkan satu pengguna tamu mengakses dan mengubah objek (berkas/file) yang sama. Pengguna terdaftar memiliki lingkungan dan ruang direktori yang terpisah dari pengguna lainnya. Sedangkan administrator memiliki kuasa untuk mengelola seluruh sistem dan aplikasi serta kode yang dibuat oleh pengguna.

Pengguna (baik pengguna tamu maupun pengguna terdaftar) dapat membuat /menuliskan/mengunnggah kode program melalui antar muka web yang digambarkan pada gambar 3. Melalui antar muka ini pengguna juga dapat mencari dan mengubah kode sumber yang sudah ada. Kode sumber yang sudah disimpan dapat dikompilasi dan dijalankan pada sistem cluster. Antar muka web juga menampilkan hasil atau pesan kesalahan yang terjadi dari kode yang dikompilasi atau dijalankan sebelumnya.

Gambar 3 Antarmuka web dari kompilator online

Berdasarkan ancaman yang dikenal sebelumnya, kami merancang dua tahapan pengamanan terhadap sistem ini. Pengamanan tahap pertama adalah pengamanan aplikasi depan (*front-end*). Pengamanan ini meliputi berbagai pertimbangan keamanan aplikasi online yang diuraikan pada bagian 2, yang meliputi:

- Validasi data masukan
  Data dari pengguna seperti nama pengguna, kata sandi, parameter kompilasi dan eksekusi, kode sumber divalidasi dengan menggunakan metode daftar hitam (*black-list*). Terdapat beberapa baris kode, parameter, dan masukan yang harus disaring sebelum dikirimkan ke sistem karena dapat merusak sistem seperti penghapusan berkas, penampilan informasi basis data, dll. Parameter yang dikirimkan harus diperiksa melalui daftar tersebut. Metode ini tidak hanya dapat melindungi dari *sql injection* atau *cross-site scripting*, tetapi juga melindungi sistem dari kode yang memanggil pustaka sistem operasi yang disisipkan dalam kode sumber yang dikirimkan.
- *Authentication*
  Sistem menggunakan otentifikasi nama pengguna dan kata sandi dimana kata

sandi disimpan pada basis data dengan menggunakan enkripsi MD5.

- *Authorization*
  Menggunakan daftar peran dan akses, dimana pengguna dikategorikan berdasarkan tiga kelompok; tamu, pengguna tedaftar dan administrator. Setiap kelompok memiliki ruang dan aksi yang disahkan.
- Manajemen konfigurasi
  Karena sistem dikembangkan sebagai aplikasi web, manajemen konfigurasi diserahkan kepada pengelolaan server web. Pengelolaan konfigurasi dilakukan oleh sistem administrator.
- Kriptografi
  Sistem menggunakan kriptografi kunci simetrik dengan menggunakan algoritma Hash MD5 untuk penyimpanan informasi pengguna dan kata sandinya. Sednagkan untuk komunikasi antar sistem menggunakan algoritma kunci asimetrik RSA.
- Pengamanan data sensitive
  Aplikasi hampir tidak memiliki data sensitive kecuali password sistem dan informasi log akses pengguna.
- Manajemen eksepsi
  Eksepsi dan error dari aplikasi antar muka web disembunyikan dari pengguna, sehingga pengguna hanya memiliki akses terhadap pesan yang seragam yang menampilkan pesan kesalahan yang diperlukan pengguna.
- Manajemen sesi
  Pengguna terdaftar dan administrator dapat beada dalam sesi yang sama dalam waktu tertentu. Kami menetapkan 30 menit sebagai waktu eksekusi terlama dari suatu program paralel yang benar yang dijalankan pada sistem.
- *Auditing* dan *logging*
  Setiap aktifitas pengguna, membuat, mengunggah, mengubah, mengkompilasi, menjalankan, mendaftarkan diri dicatat dalam suatu histori (sejarah) sistem. Selain itu kode yang dibuat, pesan kesalahan dan hasil eksekusi juga disimpan. Hal ini ditujukan untuk tujuan pengauditan jika terjadi kesalahan atau perilaku sistem yang tidak diharapkan.

Tahap pengamanan kedua (*back-end*) mempertimbangkan lebih sedikit aspek keamanan dari pada tahap pertama. Hal ini disebabkan karena sistem hanya memiliki satu antarmuka komunikasi. Sistem cluster memiliki banyak komputer yang dikelola melalui komputer utama (*master*) yang memiliki antarmuka komunikasi dengan aplikasi web. Pada sistem cluster, perintah dan aksi dilakukan pada lapisan sistem operasi, sehingga banyak pertimbangan keamanan dipindahkan ke ranah sistem operasi. Peertimbangan keamanan pada tahap kedua adalah:

- Validasi data masukan
  Meskipun kode yang diterima sudah divalidasi dengan menggunakan metode *black-list*, kode yang akan dijalankan harus dianalisis lagi dengan menggunakan perangkat pengenal (*profiler*) kode sumber.
- *Authentication* dan *Authorization*
  Setiap pengguna diberikan nama dan password pengguna sistem secara acak, sehingga pengguna tidak mengetahui metode komunikasi dan penggunaan sistem pada level sistem operasi. Komunikasi antar pengguna sistem operasi dilakukan menggunakan komunikasi terjaga (*secure communication*).
- Manajemen konfigurasi
  Manajemen konfigurasi dikelola oleh administrator cluster dan administrator sistem operasi.
- Kriptografi
  Komuniasi antar computer pada cluster menggunakan kriptografi kunci asimetrik.
- Pengamanan data sensitive
  Data sensitif diamankan dengan mengugunakan kunci asimetrik, yang meliputi log akses pengguna, dan konfigurasi sistem
- Manajemen eksepsi
  Pesan kesalahan yang ditampilkan ke pengguna hanya yang berhubungan

dengan kesalahan kompilasi dan eksekusi. Kesalahan lainnya tidak ditampilkan kepada pengguna.

- Manajemen sesi
  Manajemen sesi tidak dilakukan
- *Auditing* dan *logging*
  Pengauditan dan log dikelola oleh sistem oeprasi

Rancangan keamanan sistem ini diharapkan dapat mengamankan sistem tanpa menurunkan kebergunaan dan fungsi sistem sebagai kompilator online untuk pemrograman paralel. Kami mengembangkan nya dengan menggunakan Apache web server, php dan mysql dengan selalu memperhatikan aspek keamanan sistem.

5. Kesimpulan

Rancangan pendekatan keamanan yang kami gunakan untuk pengembangan kompilator online untuk pemrograman paralel dibagi menjadi dua mekanisme pengamanan (*front-end* dan *back-end*) yang diharapkan dapat melindungi aplikasi dari pengguna atau kode yang berbahaya. Sampai saat ini kami masih menguji pendekatan ini sebelum aplikasi digunakan pada lingkungan internet yang tidak aman. Sejauh ini, kami meperoleh hasil yang cukup baik dengan menggunakan pendekatan ini dalam mendeteksi maksud dan serangan yang mebahayakan

6. Daftar pustaka

[1] Gary McGraw, *Software Security*, Addison Wesley Professional, 2006.

[2] Michael Howard, David LeBlanc and John Viega, *19 Deadly Sins of Software Security: Programming Flaws and How to Fix Them*, McGraw-Hill/Osborne, 2005.

[3] Taufiq Wirahman, Wiwin Suwarningsih, Andria Arisal, Nuryani, *Online Compiler untuk Pembelajaran Pemrograman Paralel*, Proceeding of Seminar Riset Teknologi Informasi, Yogyakarta, 7-8 August 2009

[4] J.D. Meier, *Web Application Security Engineering*, IEEE Security and Privacy, vol. 4, no. 4, pp. 16-24, July/Aug. 2006, doi:10.1109/MSP. 2006.109

[5] Neil Daswani,Christoph Kern, and Anita Kesavan, *Foundations of Security What Every Programmer Needs to Know*, Apress, 2007.

[6] Yao-Wen Huang, Shih-Kun Huang, Tsung-Po Lin, Chung-Hung Tsai, *Web application security assessment by fault injection and behavior monitoring*, Proceedings of the 12th international conference on World Wide Web, Budapest, pp 148 - 159, 2003

[7] James B. D. Joshi, Walid G. Aref, Arif Ghafoor, Eugene H. Spafford, *Security models for web-based applications*, Communications of the ACM, Volume 44 , Issue 2 , pp 38 - 44, 2001