**STANDAR SECURE - SOFTWARE DEVELOPMENT LIFE CYCLE (S-SDLC)**

**{{nama organisasi}}**

**{{logo organisasi}}**

|  |  |
| --- | --- |
| No Dokumen | : 0.2 (Draft) |
| Revisi | : 00 |
| Tanggal Terbit | : |

|  |  |  |
| --- | --- | --- |
| Disusun  oleh : | Diketetahui  oleh: | Disetujui / Disahkan  oleh : |
|  |  |  |
| **NN**  NN | **NN**  NN | **NN**  NN |

**Riwayat Dokumen**

|  |  |  |
| --- | --- | --- |
| **Versi** | **Tanggal** | **Perubahan** |
| 0.1 (Draft) | 28/02/2023 | Rilis draft awal dengan hanya menyalin sepenuhnya konten dari **PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB - BSSN** |
| 0.2 (Draft) | 29/02/2023 | * Penambahan Diagram Tahapan S-SDLC |
|  |  |  |

1. **TUJUAN**

Memastikan bahwa pembangunan atau pengembangan sistem informasi (*Software Development Life Cycle*) berbasis web selalu memperhatikan aspek – aspek keamanan dalam setiap tahapan.

1. **RUANG LINGKUP**
   1. Tahapan umum SDLC
   2. Aspek keamanan yang diperhatikan dalam setiap tahapan SDLC
2. **ISTILAH DAN DEFINISI**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Secure Software Development Life Cycle | : | Serangkaian kegiatan yang dilakukan untuk mengembangkan, memelihara, dan memberikan solusi perangkat lunak yang aman. Kegiatan mungkin tidak harus  berurutan; mereka bisa bersamaan atau berulang. |
| 2. | Whitelist | : | Daftar sekumpulan web domain atau alamat email, dan URL yang terindikasi “aman” sehingga secara otomatis akan diterima oleh komputer maupun jaringan agar dapat diakses. Teknologi whitelist banyak digunakan karena menjadi kontrol akses sehingga komputer atau jaringan tertentu dapat mengakses domain atau menerima email yang diperbolehkan  atau tidak diperbolehkan oleh organisasi atau individu. |
| 3. | Instansi | : | Kementerian/Lembaga, Instansi pusat atau daerah. |
| 4. | Session ID | : | Nomor unik yang diberikan [server](https://www.kamuskomputer.com/definisi/server) situs Web untuk pengguna tertentu selama durasi kunjungan ([session](https://www.kamuskomputer.com/definisi/session)). ID sesi dapat disimpan sebagai cookie, form field, atau [URL](https://www.kamuskomputer.com/definisi/URL) (Uniform Resource Locator).  Beberapa server Web menghasilkan ID sesi dengan hanya menambah angka statis. Namun, sebagian besar server menggunakan [algoritm](https://www.kamuskomputer.com/definisi/algorithm)a yang melibatkan metoda yang lebih kompleks, seperti memfaktorkan dalam tanggal dan waktu kunjungan bersama dengan [variable](https://www.kamuskomputer.com/definisi/variable)s yang ditentukan oleh  administrator server |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | *Password* | : | Kata sandi yang digunakan bersamaan dengan *username*  (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer. |
| 6. | *Patch* | : | Rutin program atau sekumpulan kecil instruksi yang biasanya dibuat sebagai solusi sementara untuk mengatasi atau memperbaiki permasalahan (*bugs*) pada program komputer dan sering dibuat dalam bentuk *’object code’* yang disisipkan  ke dalam program yang akan dieksekusi. |
| 7. | *Cookie* | : | Bagian kecil data yang dikirim dari situs web dan disimpan di komputer pengguna oleh browser web pengguna saat  pengguna sedang menjelajah. |
| 8. | *Session* | : | Pertukaran informasi sementara dan interaktif antara dua atau lebih perangkat komunikasi atau antara komputer dan  pengguna |

1. **REFERENSI**
   1. OWASP Secure Software Development Life Cycle Project (S-SDLC)
2. **STANDAR**
   1. **Tahapan S-SDLC**

Timeline, calendar

Description automatically generated with medium confidence

* + 1. Pembuatan Kerangka Acuan Kerja & Perencanaan Proyek
       1. Dokumen Kerangka Acuan Kerja harus berisikan bagian yang secara jelas menyebutkan:
          - Kebutuhan fungsional keamanan aplikasi
          - Kebutuhan jaminan keamanan aplikasi
          - Kebutuhan dokumentasi terkait keamanan aplikasi
          - Deskripsi lingkungan pengembangan, pengujian, pra-produksi/staging dan produksi/operasional.
          - Syarat uji terima keamanan aplikasi
       2. Dokumen Perencanaan proyek harus berisikan bagian yang secara jelas menyebutkan:
          - Tahapan peninjauan terkait keamanan aplikasi melalui pemeriksaan dokumen formal SDLC maupun pelaksanaan skenario ujicoba khusus untuk keamanan aplikasi.
          - Ada pihak yang secara jelas ditunjuk sebagai Security Officer yang bertugas melakukan peninjauan dan audit atas pelaksanaan kegiatan diatas.
          - Bagian yang memastikan bahwa rencana pekerjaan yang berhubungan dengan pemenuhan kebutuhan maupun standar pedoman keamanan aplikasi memiliki sumber daya tenaga ahli, waktu dan dana yang cukup dan memadai.
    2. Identifikasi Kebutuhan Bisnis
       1. Dokumen kebutuhan bisnis (*Business Requirement Spesification*, BRS) wajib mencantumkan persyaratan keamanan aplikasi dari sisi bisnis pengguna.
       2. Pembagian peran (*application role*) antar pengguna harus didefinisikan dengan jelas dengan menggunakan kaidah pemisahan tugas fungsi (*segregation of duty*).
       3. Mendefinisikan sumber daya yang dibutuhkan untuk melaksanakan Standar keamanan yang diperlukan oleh aplikasi dan juga yang dibutuhkan dalam proses pengembangannya.
       4. Menerapkan manajemen risiko untuk mengidentifikasikan ancaman keamanan, kemungkinan dan dampak yang berkaitan dengan penggunaan aplikasi.
    3. Identifikasi Kebutuhan Fungsional
       1. Dokumen kebutuhan fungsional (*Functional Requirement Specification*, FRS) wajib mencantumkan bagian khusus mengenai persyaratan keamanan aplikasi.
       2. Pembagian hak akses, fungsi antar menu berbasis peran pengguna harus didefinisikan dengan jelas dengan menggunakan kaidah pemisahan fungsi tugas (segregation of duty).
       3. Mendefinisikan persyaratan ujicoba fungsional khusus untuk kebutuhan keamanan aplikasi, termasuk di dalamnya standar tool yang digunakan untuk pengembangan, untuk melakukan ujicoba keamanan, mekanisme, skenario dan standar lolos ujicoba
       4. Mendefinisikan sumber daya yang dibutuhkan untuk melaksanakan standar keamanan yang diperlukan oleh aplikasi dan juga yang dibutuhkan dalam proses pengembangannya.
       5. Menerapkan manajemen risiko untuk mengidentifikasikan ancaman keamanan, kemungkinan dan dampak yang berkaitan dengan penggunaan aplikasi dari sudut pandang teknis.
    4. Desain Aplikasi
       1. Desain aplikasi wajib mengakomodasi kebutuhan keamanan yang didefinisikan dalam dokumen BRS dan FRS. Disain aplikasi ditinjau oleh unit IT Operation untuk memastikan kesesuaian dengan kebutuhan persyaratan keamanan dalam BRS dan FRS.
       2. Desain sistem aplikasi dibuat harus dengan memperhatikan ancaman-ancaman serangan yang mungkin terjadi pada titik-titik yang digambarkan dalam arsitektur berikut ini:

**Gambar 3.7 Aspek Keamanan dalam Desain Aplikasi**

Diagram

Description automatically generated

* + 1. Konstruksi Aplikasi
       1. Pengembang harus mematuhi Standar keamanan aplikasi dan *Standar Quality Control* dalam melakukan konstruksi aplikasi.
       2. Pengembangan aplikasi harus mematuhi persyaratan keamanan yang telah ditentukan dalam dokumen disain aplikasi.
       3. Pengembang harus menggunakan Standar tool pemrograman yang telah disahkan oleh IT Development sebagai tools yang telah memiliki fungsi validasi keamanan Standar.
       4. Hasil konstruksi harus ditinjau oleh fungsi audit untuk memastikan pekerjaan yang dilakukan telah sesuai dengan persyaratan keamanan aplikasi.
    2. Pengujian Aplikasi
       1. Pengembang harus membuat skenario ujicoba khusus yang mengevaluasi persyaratan keamanan yang ditentukan dalam dokumen BRS dan FRS.
       2. Skenario ujicoba yang mengevaluasi hasil konstruksi aplikasi berdasarkan pedoman keamanan aplikasi harus dibuat dan dilaksanakan oleh fungsi audit.
       3. Skenario ujicoba berisi Standar yang harus dipenuhi untuk setiap skenario.
       4. Hasil ujicoba didokumentasikan dan ditindaklanjuti jika masih ditemukan ketidaksesuaian dengan standar yang telah ditentukan.
    3. Sosialisasi & Pelatihan Keamanan Aplikasi
       1. Pengembang aplikasi baik internal maupun rekanan wajib memperoleh pelatihan dan sosialisasi mengenai kebijakan keamanan informasi, pedoman keamanan pengembangan aplikasi, dan standar SDLC. Pelatihan dan sosialisasi yang dilakukan adalah:
          - Sosialisasi / Pelatihan awal untuk karyawan dan rekanan baru.
          - Sosialisasi / Pelatihan pengulangan berkala 1 tahun sekali.
       2. Pengguna aplikasi selain diberikan pelatihan mengenai penggunaan aplikasi juga diberikan pelatihan dasar mengenai keamanan aplikasi terutama yang berkaitan dengan hak akses/perlindungan user & password.
       3. Setiap kegiatan pelatihan keamanan pengembangan aplikasi maupun keamanan dasar aplikasi didokumentasikan dalam laporan kegiatan pelatihan.
       4. Pelatihan awal harus diberikan kepada pengembang internal ataupun rekanan sebelum melakukan pekerjaannya.
       5. Pelatihan bagi pengguna harus diberikan kepada pengguna sebelum hak akses kepada aplikasi digunakan oleh pengguna tersebut.
       6. Pelatihan harus dilakukan kembali ketika ada perubahan sistem yang mengubah cara penggunaan atau terjadi revisi dalam dokumen pedoman ini.
  1. **Pedoman Teknis Penerapan Keamanan Pada Tahapan S-SDLC**
     1. Memvalidasi data input
        1. Mengasumsikan seluruh masukan adalah malicious.
        2. Menggunakan validasi input tersentralisasi.
        3. Tidak tergantung pada validasi di sisi client tetapi juga menggunakan validasi di sisi server.
        4. Memvalidasi tipe, panjang, format dan rentang dari data.
        5. Menyusun sebuah rutin/kode program validasi secara terpusat.
        6. Menentukan karakter set yang tepat untuk semua sumber. Contohnya UTF-8
        7. Menyandikan data ke set karakter umum sebelum divalidasi
        8. Semua kegagalan validasi harus dicatatat pada input yang ditolak
        9. Bilamana sistem mendukung set karakter UTF-8 yang dikayakan, validasi dilakukan setelah proses decoding UTF-8 selesai.
        10. Melakukan validasi semua data yang disediakan client sebelum diolah, termasuk semua parameter, URL dan konten header HTTP (Seperti nama dan nilai cookie). Memastikan untuk menyertakan umpan balik otomatis dari *java script, flash* dan kode lain yang ditanam.
        11. Memverifikasi nilai-nilai header mengandung hanya karakter ASCII pada permintaan dan respon permintaan terkait.
        12. Melakukan verifikasi data dari redirection
        13. Memvalidasi semua input dengan daftar karakter yang diijinkan, ketika kondisi tersebut memungkinkan.
        14. Dalam hal terdapat karakter yang berpotensi membahayakan pada input yang harus diterima, perlu dipastikan terdapat pengendali tambahan, seperti encoding output, pengamanan proses API-API khusus dan menghitung utilitas data dalam sistem. Contoh umum karakter berbahaya seperti < > " ' % ( ) & + \ \' \".
        15. Bila rutin/kode sumber Standar validasi tidak dapat menangani input yang masuk, input terkait harus diperiksa secara terpisah.
        16. Memeriksa semua byte bernilai null.
        17. Memeriksa semua karakter baris baru (%0d, %0a, \r, \n).
        18. Memeriksa semua “titik titik garis miring karakter perubahan jalur (../ atau ..\). Dalam kondisi dimana UTF-8 yang karakternya dikayakan digunakan, alamat jalur di representasikan seperti %c0%ae%c0%ae/.
     2. Otentikasi
        1. Mengelompokan halaman situs berdasarkan akses anonim, akses teridentifikasi, dan akses terotentikasi.
        2. Menggunakan aturan password sesuai kebijakan keamanan TI.
        3. Menggunakan masa berlaku password dan penonaktifan akun user.
        4. Tidak menyimpan credentials user di database tetapi dapat menggunakan domain controller/SSO. Enkripsi saluran komunikasi ketika mengirimkan informasi username dan password menggunakan https.
     3. Otorisasi
        1. Menerapkan *whitelist* untuk *web server* dan *application server* yang bisa mengakses sql request ke database server.
        2. Halaman yang diidentifikasikan untuk akses terotentikasi selalu memvalidasi user session di server dan hak aksesnya terhadap halaman tersebut, bukan sekedar masa berlaku sesinya saja kecuali jika memang ditujukan untuk publik.
        3. Membatasi akses aplikasi ke *system level resources server* seperti files, folders, registry keys, Active Directory objects, database objects, event logs dan sebagainya.
        4. Menggunakan Windows Access List untuk membatasi user apa yang bisa mengakses resource dan operasi apa yang bisa dilakukannya.
     4. Data Sensitif/Rahasia
        1. Tidak menyimpan dan mengirimkan data user dan password database dalam plain text.
        2. Menggunakan secure communication (SSL) ketika mengirimkan user dan password database.
     5. Manajemen Sesi
        1. Membatasi masa hidup / berlaku session.
        2. Enkripsi isi dari cookies untuk otentikasi.
        3. Melindungi session state dari akses yang tak terotorisasi atau pencurian sesi.
        4. Memastikan bahwa sesi ditutup ketika pengguna log out
        5. Memastikan bahwa sesi mengalami timeout setelah tidak aktif selama jangka waktu tertentu.
        6. Memastikan bahwa semua halaman yang membutuhkan otentikasi untuk diakses, memiliki tautan untuk logout.
        7. Memastikan bahwa nomor sesi tidak pernah diungkapkan selain di header cookie, khususnya di URL, pesan error, atau log. Ini termasuk memastikan bahwa aplikasi tidak memperbolehkan penulisan ulang session cookies melalui URL.
        8. Memastikan bahwa nomor sesi tidak pernah diungkapkan selain di header cookie, khususnya di URL, pesan error, atau log. Ini termasuk memastikan bahwa aplikasi tidak memperbolehkan penulisan ulang session cookies melalui URL.
        9. Memastikan bahwa ID sesi diubah setiap kali login dan setiap otentikasi ulang.
        10. Memastikan bahwa sesi id berubah atau dibersihkan pada saat logout.
        11. Memastikan bahwa id sesi yang diakui sebagai sah oleh aplikasi adalah hanya id yang dihasilkan oleh framework aplikasi.
        12. Memastikan bahwa token sesi yang telah terotentikasi cukup panjang dan acak untuk bertahan dari jenis serangan yang sering terjadi berdasarkan ancaman ancaman di lingkungan.
     6. Manipulasi Parameter
        1. Tidak mempercayai field yang dapat dimanipulasi oleh client (query strings, form fields, cookies, atau HTTP headers).
        2. Memvalidasi semua nilai parameter yang dikirim oleh client.
     7. Exception Handling
        1. Menggunakan exception handling yang terstruktur untuk memudahkan penambahan maupun pencarian masalah.
        2. Tidak menampilkan pesan kesalahan secara detil dari server.
        3. Tidak mencatat log data pribadi seperti password.
     8. Log dan Audit
        1. Aplikasi harus memiliki fungsi audit dan log minimal pada fungsi login dan akses halaman web.
        2. Aplikasi harus dapat mencatat setiap login aplikasi yang sukses dan yang gagal, kapan, siapa usernya, dari IP berapa dan nama komputernya.
        3. Aplikasi harus bisa mencatat user, kapan, IP dan nama computer dari setiap sesi user yang mencoba mengakses halaman yang tidak diberikan hak akses untuknya.
     9. Konfigurasi Web Server
        1. Menghapus atau menonaktifkan akun yang tak digunakan
        2. Menonaktifkan Akun Guest
        3. Mengganti nama akun administrator
        4. Menonaktifkan akun IUSR (default akun internet anonim)
        5. Membuat akun Aplikasi Berbasis Webanonim sendiri secara custom.
        6. Menggunakan kebijakan password yang kuat
        7. Menonaktifkan anonymous logons.
        8. Membatasi remote logons.
        9. Menghapus virtual direktori berikut yang terinstal sebagai contoh: IISSamples, IISAdmin, IISHelp,dan Scripts.
        10. Menonaktifkan setelan Parent Path untuk menghindari serangan direktori traversal.
        11. Tidak boleh memasang protokol yang plain text dan berbahaya secara bawaan seperti: Telnet, Post Office Protocol (POP3), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP)

* + 1. Proses Persetujuan (Approval) melalui tautan email
       1. Penggunaan token unik terenkripsi untuk setiap URL permintaan persetujuan yang dikirimkan melalui suatu email yang kode parameternya tidak mudah dipahami user.
       2. Adanya konfirmasi kembali dari sistem atas data yang berhasil / gagal dicatat oleh sistem atas tindakan yang dilakukan oleh pengguna
       3. Memastikan pencatatan kontrol keamanan menyediakan kemampuan untuk mencatat peristiwa keberhasilan dan kegagalan yang diidentifikasi sebagai yang terkait keamanan.
       4. Permintaan email persetujuan tidak boleh menggunakan alamat email pemohon sebagai pengirim, harus menggunakan alamat email akun dari sistem aplikasi