

Kerentanan, Patching, dan System Hardening

Minggu 3 - 13S4121 Keamanan Siber

Baseline aman, secure configuration, manajemen perubahan, dan verifikasi

Albert Sagala, S.T., M.T.

Sarjana Teknik Elektro - Institut Teknologi Del

Posisi Materi dalam RPS

Sub-CPMK 3

Mengidentifikasi kerentanan serta menerapkan patching dan hardening dasar. [C3, C4]

Kata kunci minggu ini

- ▶ Vulnerability dan exposure
- ▶ Patching berbasis risiko
- ▶ Baseline dan secure configuration
- ▶ Configuration drift dan rollback

Orientasi belajar

- ▶ Hardening bukan pengaturan sekali jadi
- ▶ Perubahan harus terkelola dan dapat diuji
- ▶ Status “patched” bukan bukti cukup
- ▶ Setiap keputusan perlu pemilik, bukti, dan jadwal review

Materi ini merupakan versi mahasiswa dan tidak memuat kunci jawaban asesmen final.

Capaian Pembelajaran Mingguan

Pada akhir pertemuan, mahasiswa mampu:

- ▶ Membedakan kerentanan perangkat lunak, konfigurasi, identitas, dan proses.
- ▶ Menilai kebutuhan patch berdasarkan **eksposur**, **dampak**, **exploitability**, dan **ketergantungan**.
- ▶ Menyusun baseline konfigurasi aman yang dapat diuji.
- ▶ Mendokumentasikan perubahan, rollback, dan bukti verifikasi.

Indikator keberhasilan kelas

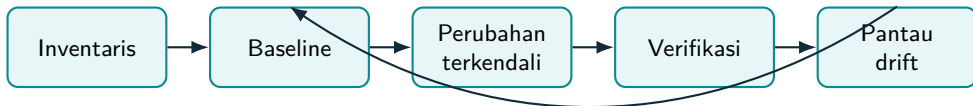
Mahasiswa dapat menyusun worksheet hardening sederhana yang berisi tujuan, baseline, prioritas perubahan, risiko perubahan, rollback, acceptance test, bukti, pemilik, dan jadwal review.

Peta Perkuliahan 100 Menit

Waktu	Aktivitas pembelajaran
0-10 menit	Apersepsi: mengapa sistem yang “berjalan normal” belum tentu aman.
10-30 menit	Konsep kerentanan, eksposur, baseline, dan secure configuration.
30-50 menit	Patching berbasis risiko: kapan diterapkan, ditunda, atau diberi kontrol kompensasi.
50-70 menit	System hardening: least functionality, akun, layanan, logging, dan verifikasi.
70-90 menit	Latihan kelompok: worksheet hardening untuk satu VM laboratorium fiktif.
90-100 menit	Rangkuman, tugas mingguan, dan pertanyaan pemahaman.

Alokasi waktu dapat disesuaikan dengan dinamika diskusi kelas.

Alur Konsep Minggu 3



Prinsip pengarah

HARDENING = reduce exposure + secure defaults + least function + verification

Hardening adalah siklus terkelola, bukan pengaturan satu kali.

Kerentanan: Apa yang Sebenarnya Rentan?

Sumber kerentanan

- ▶ Kesalahan desain atau implementasi
- ▶ Versi perangkat lunak usang
- ▶ Konfigurasi lemah
- ▶ Kredensial default
- ▶ Hak akses berlebihan
- ▶ Layanan atau fitur tidak perlu
- ▶ Proses perubahan yang buruk

Poin penting

Kerentanan tidak dapat dinilai hanya dari nama produk atau versi. Identifikasi harus menyertakan **versi**, **konteks operasi**, **eksposur**, dan **ketergantungan**.

Kerentanan = kelemahan yang dapat dimanfaatkan

Membedakan Vulnerability, Exposure, dan Risk

Konsep	Pertanyaan inti	Contoh pada server internal
Vulnerability	Apa kelemahannya?	Versi aplikasi warisan belum mendapat patch keamanan.
Exposure	Apakah kelemahan dapat dijangkau?	Port layanan hanya terbuka di jaringan internal, tetapi dapat diakses banyak segmen.
Risk	Apa skenario dampaknya?	Akses tidak sah dapat mengubah data atau mengganggu layanan organisasi.
Control	Bagaimana likelihood atau impact diubah?	Segmentasi, patch teruji, logging, backup, dan rollback.

Kalimat analisis yang baik

"Aset X memiliki kelemahan Y pada kondisi Z; jika dimanfaatkan oleh aktor A melalui jalur B, maka dapat terjadi dampak C. Kontrol D diperlukan dan efektivitasnya dibuktikan dengan E."

Patching Berbasis Risiko

Patch penting, tetapi keputusan patch perlu konteks.

- ▶ Apakah aset terpapar?
- ▶ Apakah ada eksploitasi aktif?
- ▶ Apa dampak jika patch gagal?
- ▶ Apakah patch kompatibel dengan aplikasi?
- ▶ Kapan jendela pemeliharaan tersedia?
- ▶ Apakah ada kontrol kompensasi?
- ▶ Apakah rollback telah disiapkan?

Keputusan penundaan

Penundaan patch bukan berarti mengabaikan risiko. Penundaan harus memiliki **pemilik keputusan**, **batas waktu**, **kontrol kompensasi**, dan **bukti pemantauan**.

Alur Keputusan Patch

Tahap	Pertanyaan kendali	Bukti keputusan
Identifikasi	Versi, aset, dan pemilik sudah jelas?	Inventaris aset dan versi konfigurasi.
Eksposur	Kelemahan dapat dijangkau dari mana?	Peta jaringan, daftar port, dan akses.
Prioritas	Dampak dan exploitability bagaimana?	Risk statement dan justifikasi prioritas.
Pengujian	Patch kompatibel dengan fungsi utama?	Hasil uji pada salinan atau lingkungan uji.
Penerapan	Ada jendela perubahan dan rollback?	Change log, approval, dan rencana pemulihan.
Verifikasi	Fungsi dan kontrol tetap bekerja?	Regression test, log, alert test, dan residual risk.

Pesan utama

Alur patching harus menunjukkan alasan, perubahan, pengujian, rollback, dan verifikasi; bukan hanya

Baseline dan Secure Configuration

Definisi kerja

Baseline adalah keadaan konfigurasi yang disetujui untuk suatu peran sistem dan digunakan sebagai acuan audit, perubahan, dan deteksi drift.

Elemen baseline

- ▶ Layanan dan port
- ▶ Akun dan role
- ▶ Autentikasi
- ▶ Izin akses

Kontrol operasional

- ▶ Logging
- ▶ Update policy
- ▶ Backup
- ▶ Konfigurasi jaringan

Syarat mutu

- ▶ Spesifik per peran
- ▶ Berversi
- ▶ Disetujui
- ▶ Diuji berulang

Least Functionality: Kurangi yang Tidak Perlu

Prinsip

Komponen yang tidak diperlukan memperluas attack surface dan menambah beban patching.

- ▶ Nonaktifkan layanan yang tidak mendukung tujuan sistem.
- ▶ Hapus atau nonaktifkan akun yang tidak memiliki pemilik.
- ▶ Batasi protokol dan fitur yang tidak digunakan.
- ▶ Periksa dependensi sebelum perubahan dilakukan.

Sebelum hardening

Banyak layanan aktif, banyak akun, dan banyak port terbuka membuat sistem lebih sulit diaudit.

Sesudah hardening

Hanya fungsi yang diperlukan yang tetap aktif, sehingga kontrol lebih mudah diuji dan dipertahankan.

Area Analisis dan Keputusan

Area	Risiko umum	Tindakan defensif	Bukti
Layanan	Komponen tidak perlu	Nonaktifkan setelah uji dependensi	Daftar port dan status layanan
Akun	Default atau yatim	Hapus/nonaktifkan, review pemilik	Inventaris akun dan approval
Patch	Versi rentan	Uji, terapkan, siapkan rollback	Versi dan hasil regression test
Logging	Peristiwa tak terlihat	Aktifkan event relevan dan retensi	Contoh log dan alert test

Tabel ini membantu mahasiswa mengubah rekomendasi umum menjadi kontrol yang dapat diverifikasi.

Contoh Terpandu: Server Internal

Skenario

Server internal memiliki layanan lama yang tidak digunakan dan patch tertunda karena aplikasi warisan.

Rencana yang layak

- ▶ Konfirmasi pemilik dan dependensi
- ▶ Batasi akses jaringan
- ▶ Nonaktifkan layanan yang tidak diperlukan
- ▶ Uji patch pada salinan
- ▶ Dokumentasikan rollback
- ▶ Terapkan pada jendela perubahan

Bukti setelah perubahan

- ▶ Konfigurasi sebelum-sesudah
- ▶ Status layanan dan port
- ▶ Hasil uji fungsi
- ▶ Hasil uji kontrol
- ▶ Log perubahan dan telemetry
- ▶ Residual risk

Status “patched” saja bukan bukti cukup.

Verifikasi: Kapan Perubahan Dianggap Selesai?

Perubahan belum selesai ketika:

- ▶ Paket berhasil diinstal tetapi fungsi belum diuji
- ▶ Layanan dimatikan tetapi dependensi belum dipastikan
- ▶ Logging diaktifkan tetapi alert belum diuji
- ▶ Rollback ditulis tetapi belum pernah dicoba

Selesai jika memenuhi acceptance criterion

- ▶ Fungsi utama tetap berjalan
- ▶ Kontrol keamanan bekerja
- ▶ Bukti tersedia dan dapat ditelusuri
- ▶ Residual risk diketahui
- ▶ Pemilik menerima hasil perubahan

Configuration Drift

Definisi kerja

Configuration drift adalah perubahan keadaan sistem dari baseline yang disetujui, baik karena patch, perubahan manual, otomatisasi, dependensi, maupun tindakan darurat.

Mengapa drift penting?

- ▶ Membuat audit sulit dipercaya
- ▶ Menimbulkan perbedaan antar server
- ▶ Membuka layanan atau izin yang tidak direncanakan
- ▶ Membuat troubleshooting lebih lama

Sikap defensif

- ▶ Deteksi drift secara berkala
- ▶ Tinjau penyebab perubahan
- ▶ Bedakan perubahan sah dan tidak sah
- ▶ Jangan hanya koreksi otomatis tanpa analisis

Aktivitas Laboratorium atau Simulasi

Instruksi inti

Pada VM laboratorium yang disediakan, bandingkan konfigurasi awal dengan checklist baseline. Pilih tiga perubahan berisiko rendah, catat sebelum-sesudah, lakukan uji fungsi dan uji kontrol, lalu pulihkan jika acceptance criterion gagal.

Yang dilakukan

- ▶ Gunakan VM laboratorium sesuai scope
- ▶ Catat konfigurasi awal
- ▶ Terapkan perubahan kecil dan terkontrol
- ▶ Verifikasi fungsi dan kontrol
- ▶ Dokumentasikan rollback bila gagal

Yang dilarang

- ▶ Mengakses host di luar scope
- ▶ Melakukan pemindaian pada sistem nyata
- ▶ Menggunakan data pribadi tanpa kebutuhan
- ▶ Mengabaikan stop conditions
- ▶ Memperluas percobaan tanpa izin

Bukti Minimum yang Harus Dipertahankan

- ▶ Scope, otorisasi, topologi atau setup
- ▶ Versi konfigurasi dan kondisi awal
- ▶ Raw log atau observasi dengan identifier
- ▶ Timestamp untuk setiap perubahan
- ▶ Tabel yang memisahkan bukti, asumsi, dan interpretasi
- ▶ Hasil uji positif dan negatif
- ▶ Anomali, keterbatasan, perubahan, rollback
- ▶ Residual risk dan acceptance criterion

Prinsip dokumentasi

Bukti harus membuat orang lain dapat memahami apa yang diubah, mengapa diubah, bagaimana diuji, dan bagaimana dipulihkan jika gagal.

Worksheet Hardening: Template Jawaban Mahasiswa

Bagian	Isi minimum
Tujuan sistem	Fungsi utama, pemilik, dan batas sistem.
Baseline	Layanan, port, akun, logging, update, dan konfigurasi jaringan yang disetujui.
Prioritas perubahan	Perubahan yang paling menurunkan exposure dengan risiko operasional rendah.
Risiko perubahan	Dampak jika layanan terganggu atau dependensi gagal.
Rollback	Cara kembali ke kondisi aman sebelumnya.
Acceptance test	Uji fungsi dan uji kontrol yang harus lulus.
Bukti	Screenshot, log, konfigurasi sebelum-sesudah, dan hasil uji.
Review	Pemilik, jadwal review, dan residual risk.

Checklist Interpretasi

Gunakan checklist ini sebelum menyimpulkan bahwa hardening berhasil.

- ▶ Apakah tujuan, aset, scope, dan pemilik disebutkan?
- ▶ Apa yang merupakan fakta, hasil ukur, dokumentasi, inferensi, atau asumsi?
- ▶ Bagaimana kontrol mengubah likelihood atau impact?
- ▶ Bukti apa yang menguji efektivitas dan residual risk?
- ▶ Apakah rekomendasi memiliki owner, prioritas, dan acceptance test?
- ▶ Apakah rahasia dan data pribadi tidak dimasukkan tanpa kebutuhan?

Kesimpulan keamanan harus dapat ditelusuri dari bukti, bukan dari asumsi.

Tugas Mingguan

Tugas 3

Susun worksheet hardening untuk satu peran sistem fiktif.

Worksheet harus memuat:

- ▶ Tujuan sistem
- ▶ Baseline
- ▶ Prioritas perubahan
- ▶ Risiko perubahan
- ▶ Rollback
- ▶ Acceptance test
- ▶ Bukti
- ▶ Pemilik
- ▶ Jadwal review

Gunakan sistem fiktif agar analisis fokus pada metode, bukan pada akses ke sistem nyata.

Pertanyaan Pemeriksaan Diri

1. Mengapa patch terbaru belum tentu langsung diterapkan?
2. Apa bedanya hardening dan patching?
3. Bagaimana baseline membantu audit?
4. Mengapa rollback perlu diuji?
5. Bukti apa yang menunjukkan kontrol tetap berfungsi?

Diskusi singkat

Jawaban yang baik menyebutkan konteks, risiko operasional, bukti verifikasi, dan residual risk.

Rangkuman Satu Slide

1. Identifikasi	2. Kendalikan	3. Verifikasi
Mulai dari inventaris, versi, exposure, pemilik, dan konteks operasi.	Gunakan baseline, least functionality, patch teruji, dan kontrol kompensasi.	Simpan bukti fungsi, kontrol, rollback, telemetry, dan residual risk.

Hardening yang baik selalu dapat diuji, ditelusuri, dan dipulihkan.

Referensi dan Bacaan Lanjutan



Stallings, W. *Cryptography and Network Security: Principles and Practice*. Pearson.



Easttom, C. *Computer Security Fundamentals*. Pearson.



NIST. *Cybersecurity Framework (CSF) 2.0*. <https://www.nist.gov/cyberframework>



NIST. *SP 800-40 Rev. 4: Enterprise Patch Management Planning*.
<https://csrc.nist.gov/pubs/sp/800/40/r4/final>



OWASP. *OWASP Top 10:2025 - Security Misconfiguration*.
https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/

Catat judul, versi, tanggal akses, parameter, serta lingkungan yang digunakan ketika menggunakan standar, katalog kerentanan, dokumentasi vendor, atau threat intelligence.

Akhir Materi Minggu 3

Kerentanan, Patching, dan System Hardening

Pertanyaan dan diskusi