MODUL PERKULIAHAN:
KEAMANAN SIBER

Penyusun 1	Penyusun 2	Penyusun 3
Denpasar, <*Arial, 9pt>	Denpasar, <*Arial, 9pt>	Denpasar,
(Nama Dosen dan gelar) <*Arial, 9pt>	(Nama Dosen dan gelar) <*Arial, 9pt>	Nama Dosen dan gelar) <*Arial, 9pt>

INFORMASI MATA KULIAH

Jumlah SKS	**4**
Jenis Matakuliah	Teori
Kegiatan Pembelajaran	Daring
Persyaratan Sarana Peserta/Spesifikasi/ Tools/Media ajar yang akan digunakan	

CAPAIAN DAN DESKRIPSI MK

Capaian Pembelajaran Mata Kuliah (CPMK)	Deskripsi Singkat MK
• CPMK-06-17: Mampu menjelaskan dan menguasai teknik pengamanan sistem dan jaringan komputer untuk mengidentifikasi dan mencegah serangan cyber dengan mengimplementasikan aspek-aspek keamanan informasi. • CPMK-07-4: Mampu menjelaskan dan menguasai konsep dasar teori jaringan mahasiswa mampu memahami konsep jaringan komputer, mengetahui perangkat-perangkat jaringan, protokol jaringan, OSI dan TCP/IP model, konsep IP Address dan kelas-kelas dalam IP Address, Routing dengan media virtual seperti menggunakan packet tracer, memahami jaringan nirkabel, mengetahui serangan dalam jaringan dan mengetahui firewall.	Keamanan siber adalah matakuliah yang mempelajari tentang konsep dan teknik untuk pengamanan aplikasi, sistem dan jaringan komputer serta data digital dari kejahatan cyber yang dapat menggangu proses bisnis dan mengakibatkan kerugian.

REFERENSI

Referensi
[1] Sadikin, R. (2012). Kriptografi Untuk Keamanan Jaringan dan Implementasinya dalam bahasa java. Yogyakarta: Andi
[2] Weidman, G. (2014). Penetration Testing A Hands-On Introduction to Hacking.
[3] Situmeang, M.S (2020). Cyber Law, Bandung: Cakra 2020

Hardening System

Capaian Pembelajaran Mata Kuliah
• CPMK-06-17: Mampu menjelaskan dan menguasai teknik pengamanan sistem dan jaringan komputer untuk mengidentifikasi dan mencegah serangan cyber dengan mengimplementasikan aspek-aspek keamanan informasi.

Indikator Penilaian
1.1. Pengenalan keamanan sistem 1.2 Teknik hardening 1.2. Windows dan Linux Hardening 1.3. Patch management 1.4 Network Access Quarantine Control 1.5 Security Auditing and Event Logs

BAB 1: Konsep Dasar Hardening System

Pengertian dan tujuan hardening system

Hardening system adalah proses memperkuat keamanan pada suatu sistem komputer, server, atau jaringan untuk melindungi dari ancaman siber. Proses ini melibatkan pengaturan konfigurasi dan penonaktifan fitur atau layanan yang tidak dibutuhkan, serta peningkatan keamanan melalui penambahan lapisan perlindungan dan pengawasan. *Hardening* juga mencakup langkah-langkah seperti pembaruan perangkat lunak, pembatasan akses, penggunaan enkripsi, dan penerapan kebijakan keamanan yang ketat.

Tujuan Hardening System

1. **Mengurangi Kerentanan**
   Memperkecil potensi kerentanan yang dapat dieksploitasi oleh atacker atau perangkat lunak berbahaya.
2. **Mencegah Akses yang Tidak Diinginkan**
   Membatasi akses hanya kepada user yang memiliki otorisasi, sehingga sistem lebih aman dari intruder.
3. **Meningkatkan Keamanan Data**
   Melindungi data sensitif dari akses atau perubahan yang tidak sah.
4. **Meminimalkan Risiko Serangan**
   Membuat sistem lebih sulit diserang melalui *exploits* atau metode lain yang memanfaatkan celah keamanan.
5. **Memastikan Ketersediaan Sistem**
   Dengan sistem yang aman, risiko kegagalan operasional akibat serangan atau kesalahan pengguna dapat dikurangi, sehingga sistem tetap dapat berjalan dengan lancar.

Jenis-jenis kerentanan sistem

1. **Kerentanan Jaringan (Network Vulnerabilities)**
   Kerentanan ini muncul dari konfigurasi atau celah dalam jaringan, seperti penggunaan port yang tidak aman, pengaturan firewall yang lemah, atau protokol jaringan yang rentan. Contoh: serangan *Man-in-the-Middle* (MitM), *IP Spoofing*, dan serangan pada port terbuka.
2. **Kerentanan Sistem Operasi (Operating System Vulnerabilities)**
   Sistem operasi sering memiliki kerentanan karena bug, konfigurasi default yang tidak aman, atau kekurangan dalam proses otentikasi. Celah ini dapat memungkinkan serangan pada tingkat *root* atau kernel.
3. **Kerentanan Aplikasi (Application Vulnerabilities)**
   Kerentanan ini terjadi ketika aplikasi memiliki bug atau celah yang memungkinkan penyerang untuk menyisipkan kode berbahaya. Contoh: SQL Injection, Cross-Site Scripting (XSS), dan Cross-Site Request Forgery (CSRF).
4. **Kerentanan Konfigurasi (Configuration Vulnerabilities)**
   Kesalahan konfigurasi, seperti penggunaan pengaturan default, akses admin tanpa batas, atau tidak adanya pembatasan akses, dapat menciptakan titik lemah yang mudah dieksploitasi.

5. **Kerentanan Fisik (Physical Vulnerabilities)**
   Kelemahan dalam kontrol fisik yang memungkinkan akses langsung ke perangkat keras atau server. Contoh: akses ke server yang tidak diamankan, atau perangkat penyimpanan yang tidak terlindungi.
6. **Kerentanan Manusia (Human Vulnerabilities)**
   Faktor manusia, seperti kesalahan pengguna atau tidak adanya pelatihan keamanan, dapat menyebabkan celah keamanan. Contoh: penggunaan kata sandi yang lemah, penipuan phishing, atau kelalaian dalam pengelolaan data sensitif.
7. **Kerentanan Protokol Keamanan (Security Protocol Vulnerabilities)**
   Kekurangan atau kelemahan dalam protokol keamanan, seperti SSL/TLS yang lemah atau autentikasi yang tidak efektif, yang memungkinkan serangan seperti *SSL stripping* atau dekripsi lalu lintas.
8. **Kerentanan dalam Prosedur dan Kebijakan (Policy and Procedure Vulnerabilities)**
   Kebijakan keamanan yang lemah atau prosedur yang tidak ketat dapat menyebabkan sistem rentan. Contoh: kebijakan pembaruan perangkat lunak yang lambat, atau tidak adanya pemantauan aktivitas pengguna.

BAB 2: Pengamanan Akses dan Autentikasi

Autentikasi berbasis password

Autentikasi berbasis password adalah metode pengamanan di mana pengguna harus memasukkan password yang sesuai dengan identitas untuk mengakses sistem, aplikasi, atau layanan tertentu. Password berfungsi sebagai mekanisme verifikasi yang memastikan bahwa hanya pengguna yang mengetahui password tersebut yang dapat masuk ke dalam sistem.

Cara Kerja

Dalam proses autentikasi ini, pengguna diminta untuk:

1. Memasukkan identitas pengguna atau username.
2. Memasukkan password yang telah mereka buat sebelumnya.

Sistem akan memeriksa kecocokan password dengan data yang ada dalam basis data. Jika sesuai, akses akan diberikan. Jika tidak, akses akan ditolak.

Kelebihan dan Kekurangan

- **Kelebihan**
  Metode ini sederhana dan mudah diterapkan, serta familiar bagi kebanyakan pengguna.
- **Kekurangan**
  Rentan terhadap pencurian atau serangan brute-force. Jika password lemah atau mudah ditebak, keamanan sistem menjadi terancam.

Untuk meningkatkan keamanan autentikasi berbasis password, beberapa praktik yang dapat diterapkan meliputi :

- Menggunakan password yang kuat dan unik.
- Menambahkan autentikasi dua faktor (2FA) untuk lapisan keamanan tambahan.
- Mengganti password secara berkala dan menghindari penggunaan ulang.

Autentikasi berbasis password tetap menjadi metode autentikasi yang umum digunakan, meskipun banyak sistem mulai beralih ke metode autentikasi yang lebih aman, seperti autentikasi biometrik atau berbasis token.

BAB 3: Hardening Sistem Operasi Linux

Update sistem dan manajemen paket

Update sistem dan manajemen paket adalah proses penting dalam pengelolaan perangkat lunak dan keamanan sistem operasi. Update sistem bertujuan untuk memastikan perangkat lunak selalu mendapatkan versi terbaru, termasuk perbaikan keamanan, peningkatan fitur, dan perbaikan bug. Manajemen paket membantu dalam penginstalan, pemeliharaan, dan pembaruan paket perangkat lunak pada sistem operasi.

1. Pembaruan Sistem

- **Tujuan**
  Menjaga sistem tetap aman dan stabil. Pembaruan ini sering mencakup perbaikan kerentanan keamanan yang ditemukan setelah rilis perangkat lunak awal.
- **Jenis Pembaruan:**
  a. Update Keamanan yaitu Mengatasi kerentanan yang dapat dieksploitasi oleh peretas.
  b. Update Fitur yaitu Menambah atau meningkatkan fitur.
  c. Update Kinerja yaitu Meningkatkan kecepatan atau stabilitas sistem.
- Cara Kerja
  Pada sistem Linux, Update dapat dilakukan melalui terminal dengan perintah seperti `sudo apt update` di distribusi berbasis Debian/Ubuntu atau `sudo yum update` pada Red Hat/CentOS.

Pengaturan kontrol akses dan hak pengguna

Kontrol akses dan hak pengguna dalam sistem operasi, terutama di lingkungan berbasis Unix/Linux adalah bagian integral dari keamanan sistem. Ini melibatkan pengaturan siapa yang dapat mengakses, mengubah, atau menjalankan file dan sumber daya dalam sistem.

1. Kontrol Akses

Kontrol akses adalah proses untuk menentukan dan memverifikasi hak pengguna untuk mengakses berbagai sumber daya. Terdapat beberapa pendekatan untuk pengaturan ini:

- **Discretionary Access Control (DAC)**
  Mengizinkan pemilik file atau sumber daya untuk menentukan hak akses. Di Linux, setiap file memiliki pemilik, grup, dan izin yang dapat diubah menggunakan perintah seperti chmod, chown, dan chgrp.
- **Mandatory Access Control (MAC)**
  Mengatur akses pada tingkat kebijakan yang ditetapkan oleh administrator sistem, seperti pada sistem dengan SELinux atau AppArmor. Kontrol ini memberikan kendali lebih tinggi atas apa yang dapat dilakukan oleh pengguna dan proses.
- **Role-Based Access Control (RBAC)**
  Memanfaatkan peran yang ditetapkan untuk pengguna dalam menentukan akses, di mana setiap peran memiliki izin spesifik. Metode ini biasa digunakan pada lingkungan server yang lebih kompleks.

2. Hak Pengguna

Setiap file atau direktori di Linux memiliki tiga jenis hak akses yang menentukan tindakan yang diizinkan untuk pengguna tertentu:

- **Read (r)**
  Memungkinkan pengguna untuk melihat isi file atau direktori.
- **Write (w)**

Mengizinkan pengguna untuk mengubah atau menghapus file atau menambah isi direktori.

- **Execute (x)**
  Memungkinkan pengguna menjalankan file atau menavigasi ke dalam direktori.
  Hak akses ini dikelompokkan untuk tiga kategori pengguna:
- **Pemilik (Owner)**
  Biasanya, pengguna yang membuat file atau direktori.
- **Grup (Group)**
  Kelompok pengguna yang dapat diberikan hak akses yang sama.
- **Lainnya (Others)**
  Pengguna di luar pemilik atau grup.

3. Mengelola Kontrol Akses dan Hak Pengguna

Mengatur hak akses dapat dilakukan melalui beberapa perintah:

- **Chmod**
  Mengubah izin file atau direktori. Contoh, chmod 755 file.txt akan memberikan izin penuh kepada pemilik dan izin baca/eksekusi kepada grup dan lainnya.
- **Chown**
  Mengubah kepemilikan file atau direktori. Contoh, chown user1 file.txt akan menetapkan pengguna user1 sebagai pemilik file.
- **Chgrp**
  Mengubah grup kepemilikan. Contoh, chgrp group1 file.txt akan menetapkan group1 sebagai grup yang memiliki file.

4. Penerapan Kontrol Akses Lanjutan:

Sistem operasi juga memungkinkan penggunaan akses kontrol lanjutan melalui ACL (Access Control Lists), yang memungkinkan penentuan izin untuk pengguna dan grup tertentu secara lebih fleksibel daripada pengaturan standar.

Contoh:

Misalnya, untuk mengatur file agar hanya dapat diakses oleh pemiliknya saja, bisa menggunakan perintah `chmod 700 file.txt`

Pemantauan log dan audit sistem

Pemantauan log dan audit sistem adalah langkah penting dalam manajemen keamanan sistem untuk memantau aktivitas yang terjadi di sistem operasi dan aplikasi, serta mendeteksi ancaman atau aktivitas mencurigakan.

1. Log Sistem (System Logging)

Log adalah catatan yang disimpan oleh sistem operasi atau aplikasi terkait aktivitas pengguna, proses, kesalahan, dan kejadian lain yang terjadi. Log ini mencatat aktivitas secara terperinci, seperti akses file, perintah yang dijalankan, atau kesalahan sistem.

Jenis Log Umum:

- **Log Sistem**
  Mencatat aktivitas umum, seperti login, error, dan shutdown.
- **Log Keamanan**
  Berisi informasi tentang peristiwa keamanan, seperti login yang gagal dan perubahan hak akses.
- **Log Aplikasi**
  Mencatat aktivitas spesifik dari aplikasi tertentu, seperti server web Apache atau database.

Contoh Pemantauan Log di Linux:

- *Syslog* adalah komponen penting di Linux untuk menyimpan log sistem di `/var/log/syslog` atau `/var/log/messages`.
- *Journald* pada distribusi Linux berbasis `systemd`, menyimpan log di `/var/log/journal/`.
- *Auditd* digunakan untuk log keamanan di Linux, dengan log disimpan di `/var/log/audit/audit.log`.

BAB 4: Keamanan Jaringan dan Firewall

Konfigurasi firewall dengan ufw

Firewall Linux dapat dikonfigurasi menggunakan beberapa alat, dengan UFW (Uncomplicated Firewall) dan iptables menjadi dua yang paling umum. Berikut adalah panduan dasar untuk mengonfigurasi firewall menggunakan kedua alat ini.

1. Konfigurasi Firewall dengan UFW

UFW adalah antarmuka yang lebih mudah digunakan untuk iptables. Ini dirancang untuk menyederhanakan proses konfigurasi firewall di sistem berbasis Debian dan Ubuntu.

a. Instalasi UFW

Berikut command line untuk menginstal UFW adalah :

```
sudo apt update
sudo apt install ufw
```

b. Mengaktifkan UFW

Untuk mengaktifkan UFW menggunakan command line berikut :

```
sudo ufw enable
```

c. Menampilkan Status UFW

Untuk memeriksa status UFW menggunakan command line berikut :

```
sudo ufw status verbose
```

d. Menambahkan Aturan

Untuk mengizinkan atau menolak akses, Berikut adalah beberapa command line ya ng digunakan :

e. Mengizinkan akses SSH (port 22) :

```
sudo ufw allow ssh
```

f. Mengizinkan akses pada port tertentu (misalnya, HTTP pada port 80):

```
sudo ufw allow 80/tcp
```

g. Menolak akses pada port tertentu (misalnya, FTP pada port 21)

```
sudo ufw deny 21/tcp
```

h. Menghapus Aturan

Untuk menghapus aturan dapat menggunakan command line berikut :

```
sudo ufw delete allow 80/tcp
```

i. Menonaktifkan UFW

Untuk menonaktifkan UFW menggunakan command line berikut :

```
sudo ufw disable
```

BAB 5: Penerapan Keamanan Aplikasi

Pengamanan aplikasi dari SQL injection, XSS, dan CSRF
Keamanan aplikasi web sangat penting untuk melindungi data dan mencegah serangan yang umum seperti **SQL Injection**, **Cross-Site Scripting (XSS)**, dan **Cross-Site Request Forgery (CSRF)**. Berikut adalah penjelasan tentang masing-masing jenis serangan dan langkah-langkah untuk mengamankan aplikasi dari serangan tersebut:

1. SQL Injection

SQL Injection adalah jenis serangan yang memungkinkan penyerang untuk memasukkan kode SQL berbahaya ke dalam kueri yang dijalankan oleh aplikasi, yang dapat mengakibatkan pencurian data, modifikasi data, atau bahkan penghapusan data.

Langkah-langkah Pencegahan:

- **Menggunakan Prepared Statements dan Parameterized Queries**
- Menggunakan prepared statements dan parameterized queries untuk memisahkan kode SQL dari data yang dimasukkan oleh pengguna. Contoh dalam PHP menggunakan PDO yaitu :

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE username =
:username");
$stmt->execute(['username' => $username]);
```

2. Cross-Site Scripting (XSS)

XSS adalah serangan yang memungkinkan penyerang untuk menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dijalankan di browser pengguna. Ini bisa digunakan untuk mencuri cookie, sesi, atau data sensitif lainnya.
Langkah-langkah Pencegahan:

- **Validasi Input**
  Validasi data yang diterima dari pengguna, ini termasuk menghapus karakter berbahaya atau mengekspresikan kembali karakter khusus.
- **Menggunakan Content Security Policy (CSP)**
  Menerapkan CSP untuk membatasi sumber daya yang dapat dimuat oleh aplikasi, sehingga mengurangi risiko eksekusi script berbahaya.
- **Encode Output**
  Melakukan encode output yang ditampilkan di halaman web. Misalnya, menggunakan HTML encoding untuk data yang akan ditampilkan di browser.

3. Cross-Site Request Forgery (CSRF)

CSRF adalah serangan yang memanfaatkan kepercayaan pengguna pada aplikasi web. Dengan serangan ini, penyerang dapat mengirimkan permintaan tidak sah atas nama pengguna yang terautentikasi tanpa sepengetahuan pengguna.

Langkah-langkah Pencegahan:

Token CSRF

menggunakan token CSRF yang unik dan tidak dapat diprediksi untuk setiap permintaan yang berisiko. Dimana token ini divalidasi di sisi server sebelum memproses permintaan.

Contoh implementasi dalam PHP yaitu :

```
// Generate token
$_SESSION['csrf_token'] = bin2hex(random_bytes(32));
// Include token in forms
echo '<input type="hidden" name="csrf_token" value="' .
$_SESSION['csrf_token'] . '">';
// Validate token on form submission
if (hash_equals($_SESSION['csrf_token'],
$_POST['csrf_token'])) {
    // Process form
}
```

BAB 6: Konfigurasi dan Pengamanan SSH

SSH (*Secure Shell*) merupakan protokol yang umum digunakan untuk mengakses server secara aman. Walapun aman, SSH tetap membutuhkan konfigurasi lebih lanjut sehingga tidak mudah diserang. Beberapa langkah pengamanan SSH meliputi:

- Mengubah port default (22) untuk menghindari deteksi otomatis.
- Menonaktifkan *root login* sehingga hanya pengguna dengan kredensial tertentu yang dapat mengakses server.
- Menggunakan *key-based authentication* untuk meningkatkan keamanan login.

Contoh Konfigurasi untuk Mengubah Port SSH:

```
sudo nano /etc/ssh/sshd_config
```

Ubah Port 22 ke nomor port lain, misal 2222 Port 2222

Latihan

1. Terapkan *key-based authentication* di server Linux.
2. Ubah port SSH dan nonaktifkan login root langsung.
3. Atur firewall untuk hanya mengizinkan akses dari IP tertentu ke port SSH.