Task 3: OWASP Top 10 Summary (with examples)

A01:2021 — Broken Access Control

Access control decides who can access what. If it is broken, users can do actions they should not be able to.

How it works:
The server is supposed to check whether the user has permission to perform an action. If these checks are missing or incorrect, attackers can bypass them.

How it can happen:

- A normal user accessing admin pages
- Changing a URL to view other users' private data
- Forcefully changing account roles

Example: Facebook Business Manager (2021)

In 2021, a researcher found that by manipulating the business_id parameter in Facebook Business Manager API, they could access businesses they weren't authorized to manage.

How it was countered:

Facebook fixed the issue by enforcing server-side authorization checks and validating the business_id against the user's actual permissions.

How to avoid:

- Always check access on the server, not just on the UI
- Implement "least privilege" access
- Deny access by default and explicitly allow only required roles

A02:2021 — Cryptographic Failures

Task 3: OWASP Top 10 Summary (with examples)

Previously called Sensitive Data Exposure. The focus is now on the actual cryptographic problems, not just the symptom of leaked data.

How it works:
If data is not encrypted properly, attackers can read or modify it.
Encryption is also needed for data being sent over the internet.

How it can happen:

- Not using HTTPS
- Weak encryption algorithms (like MD5)
- Storing passwords without hashing

Example: TLS downgrade in Microsoft Exchange (2022)
An attacker could force Microsoft Exchange servers to downgrade TLS connections, making data interception possible.

How it was countered:
Microsoft patched the servers to enforce strict TLS settings and disabled insecure fallback mechanisms.

How to avoid:

- Use modern cryptography (AES, TLS 1.3)
- Hash passwords with salt (bcrypt, Argon2)
- Do not reinvent encryption—use trusted libraries

A03:2021 — Injection

Injection happens when the app accepts user input and runs it as part of a command or query.

How it works:
If user data is not validated, attackers can insert malicious code into database queries, commands, or scripts.

How it can happen:

Task 3: OWASP Top 10 Summary (with examples)

- SQL Injection (running SQL queries injected by the user)
- Command Injection (running system commands injected by the user)
- Cross-Site Scripting (XSS), now included in this category

Example: SolarWinds SQL Injection vulnerability (2021)
During the SolarWinds supply chain investigation, SQL injection flaws were found in their Serv-U FTP software.

How it was countered:
SolarWinds issued patches that replaced dynamic SQL queries with parameterized queries to block injection attempts.

How to avoid:

- Use parameterized queries (prepared statements)
- Validate and sanitize all user inputs
- Escape outputs shown on the screen (for XSS)

A04:2021 — Insecure Design

A new category in 2021. This is about flaws in the system's architecture and design, not just code bugs.

How it works:
If security is not part of the design, adding it later is very hard. Many vulnerabilities come from bad or missing design decisions.

How it can happen:

- No threat modeling during design
- No secure defaults
- Over-relying on client-side controls

Example: Zoom default settings (2020-2021)
Early in the pandemic, Zoom allowed anyone with a meeting link to join without

sufficient controls (no enforced passwords or waiting room). This led to "Zoom bombing."

How it was countered:
Zoom redesigned its platform defaults:

- *Passwords were required*
- *Waiting rooms were enabled*
- *Only hosts could allow users in*

How to avoid:

- Perform threat modeling early
- Use secure design patterns
- Build security in from the start, not as an afterthought

A05:2021 — Security Misconfiguration

Applications often have many settings. If these are not configured securely, attackers can take advantage.

How it works:
Developers or system admins may leave default settings, open ports, or detailed error messages exposed.

How it can happen:

- Leaving admin interfaces open to everyone
- Detailed error messages revealing internal workings
- Unnecessary services running

Example: Microsoft Power Apps misconfiguration (2021)
Many companies using Microsoft Power Apps exposed 38 million sensitive records due to a misconfigured feature that made APIs public.

How it was countered:
Microsoft changed the default settings to make APIs private by default and provided

guidance to all customers to review and secure their configurations.

How to avoid:

- Harden all configurations
- Disable unnecessary features and services
- Regularly test and review settings

A06:2021 — Vulnerable and Outdated Components

Using third-party software is common. But if those components are outdated or have known vulnerabilities, they can be exploited.

How it works:
Attackers look for apps using old versions of software with known flaws.

How it can happen:

- Old libraries with known CVEs (Common Vulnerabilities and Exposures)
- Not updating frameworks (e.g., old Spring, Django versions)

Example: Log4j vulnerability (Log4Shell, 2021)
The Apache Log4j library had a remote code execution vulnerability (CVE-2021-44228). Millions of applications used it.

How it was countered:
Organizations globally performed emergency updates to Log4j (versions 2.16.0+), applied temporary WAF rules, and scanned their apps for vulnerable instances.

How to avoid:

- Keep an inventory of all components
- Regularly update dependencies
- Use tools like Dependabot to find outdated packages

Task 3: OWASP Top 10 Summary (with examples)

A07:2021 — Identification and Authentication Failures

Previously called Broken Authentication. Now it also includes problems with identifying users.

How it works:
If users can pretend to be someone else or bypass login, the entire app is at risk.

How it can happen:

- Weak passwords
- Missing Multi-Factor Authentication (MFA)
- Session hijacking or fixation

Example: Twitter API bug (2022)
An API flaw allowed attackers to check whether email addresses or phone numbers were linked to Twitter accounts (user enumeration).

How it was countered:
Twitter patched the API to enforce rate limiting and changed how it responded to unauthorized requests to prevent user enumeration.

How to avoid:

- Enforce strong password policies
- Use MFA wherever possible
- Secure session tokens and timeouts

A08:2021 — Software and Data Integrity Failures

A new category in 2021. It focuses on assumptions about software updates, data, and CI/CD pipelines.

How it works:
If updates or critical data are not checked for integrity, attackers can inject malicious code.

How it can happen:

- Installing unsigned software updates
- Using untrusted plugins
- Insecure CI/CD pipelines

Example: SolarWinds supply chain attack (2020)
Attackers compromised the build system of SolarWinds Orion software and injected a backdoor into software updates.

How it was countered:

- *SolarWinds rebuilt its CI/CD pipeline with stronger code-signing and verification*
- *They enforced stricter access controls on their build systems*
- *Customers were advised to verify software integrity and apply patches.*

How to avoid:

- Sign and verify all code and updates
- Use trusted sources for components
- Secure CI/CD pipelines

A09:2021 — Security Logging and Monitoring Failures

Previously called Insufficient Logging & Monitoring. Logging helps detect attacks. If this fails, attackers can stay hidden.

How it works:
Without good logs, detecting attacks or performing forensic analysis is hard.

How it can happen:

- No logging of failed login attempts

- Logs missing key security events

- No alerts for suspicious behavior

Task 3: OWASP Top 10 Summary (with examples)

Example: Colonial Pipeline ransomware attack (2021)
Colonial Pipeline lacked sufficient monitoring and alerting, which delayed detection of the ransomware spread.

How it was countered:

- *They upgraded their logging and monitoring systems*
- *Implemented real-time alerting and centralized log collection*
- *Conducted threat hunting to proactively detect future threats.*

How to avoid:

- Log important security-related events
- Set up monitoring and alerts
- Periodically test logging and incident response

A10:2021 — Server-Side Request Forgery (SSRF)

A new entry. SSRF occurs when the server is tricked into making requests on behalf of the attacker.

How it works:

An attacker can make the server send requests to internal systems that are not exposed publicly.

How it can happen:

- Forcing an image upload feature to fetch from internal URLs
- Accessing cloud metadata services (AWS EC2 metadata)

Example: AWS metadata SSRF via Capital One breach (2019, still relevant today)
An SSRF vulnerability in Capital One's WAF allowed an attacker to access AWS EC2 instance metadata, stealing sensitive credentials.

How it was countered:

- *AWS hardened metadata services with v2 IMDS (requiring session tokens)*

Task 3: OWASP Top 10 Summary (with examples)

- *Capital One reviewed its request validation*
- *Organizations globally began applying network segmentation and SSRF protections.*

How to avoid:

- Validate and sanitize URLs used by the server
- Implement allowlists for external requests
- Use network segmentation to isolate internal systems

Case Study: Not_Petya cyberattack on Ukraine (~ $10 billion in damages)

OWASP Top 10 categories involved:

1) A08: Software and Data Integrity Failures
   → The compromised software update was not verified properly.
   → M.E.Doc's update server was not secured, allowing attackers to inject malicious code.
2) A06: Vulnerable and Outdated Components
   → EternalBlue was used because many systems had not patched the SMB vulnerability (CVE-2017-0144).
   → Even though Microsoft had issued patches, many companies hadn't applied them.
3) A09: Security Logging and Monitoring Failures
   → Many networks were blind to the lateral movement of the malware until it was too late.
   → Lack of visibility and alerting allowed rapid spread.

What happened

- Attackers compromised the update mechanism of a popular Ukrainian tax software called M.E.Doc.
- They injected malware disguised as a legitimate software update.

Task 3: OWASP Top 10 Summary (with examples)

- Once installed, NotPetya spread rapidly inside corporate networks using vulnerabilities such as EternalBlue (an SMB vulnerability leaked from the NSA toolkit)
- It encrypted systems and wiped data, causing damage.

How it was countered:

- Organizations disconnected affected networks (physical isolation).
- Emergency patching of SMB vulnerabilities.
- Hardened software update processes (signed updates, verified sources).
- Improved network segmentation to stop lateral spread.
- Enhanced logging and monitoring for future attack detection.

Sources:

1) *F5 DevCentral Community on YouTube*
2) *fern on YouTube*
3) *Google*