Tiltak i henhold til OWAST top 10 for egen server

* [**A01:2021-Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/) moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.

A01:2021 Broken Access Control fiks:

* Implementasjon av en robust tilgangs kontroll mekanisme som identifiserer brukeren
* Bruk av rollebasert tilgangskontroll som (RBAC) slik at kun de som skal ha tilgang faktisk får tilgang til spesifikke ressurser.
* Bruk av sterk passord regler og få brukerene til å ha sterk passord som er vanskelig å gjette
* Implementering av multi faktor autentisering for å få mere sikkerhet
* Å bruke kryptering for å beskytte sensitive data under overføring.
* Implementasjonen av sikker og jevnlig kodegjennomgang for å finne og fikse potensielle sårbarheter.
* Parametriserte spørringer for å forhindre SQL-Injection angrep.
* Å gjennomføre regelmessige sårbarhetsvurderinger og penetrasjonstesting for å finne og fikse sikkerhetssvakheter.
* Begrense eksponering av sensitiv informasjon ved å bruke tilgangskontroller for å begrense tilgangen til data som kun de som trenger det skal ha tilgang til.
* Monitor access logs and other security-related events to identify suspicious activity and respond quickly to potential security incidents.
* [**A02:2021-Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/) shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.

A02:2021 Cryptographic failures fiks :

* Med bruk av kryptografiske algoritmer som er anbefalt av eksperter.
* Vær sikker på at de krypterte nøklene som genereres faktisk er lagret sikkert og ikke er tilgjengelig for brukere som ikke burde ha tilgang .
* Bruk av sikker nøkkeladministrasjon praksis for å sikre at nøklene roteres jevnlig og at komprimerte nøkler blir fjernet .
* Bruk av riktig kilder for å generere krypteringsnøkler for å sikre at de er sikre nok .
* Bruk av kryptering når data blir overført for å beskytte dataen når den blir overført over nett. slik som SSL / TLS
* Bruk av kryptering når data er i hvile for å beskytte data som er lagret på databaser eller lokalt på disk.
* Bruk av sikre protokoller og biblioteker som har blitt testet for sikkerhet sårbarheter.
* Jevnlige sårbarhetsvurderinger og penetrasjonstesting for å finne og løse feil med kryptering
* Bruk av digitale signaturer for å sikre autentisitet for å hindre uautorisert tilgang og modifikasjon.
* [**A03:2021-Injection**](https://owasp.org/Top10/A03_2021-Injection/) slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.

The A03:2021 Injection fiks:

* Bruk av parametriserte spørringer for å forhindre SQL Injections .
* Bruk av input validering og filtrering for å forhindre (XSS) cross site scripting og kommando injeksjoner.
* Bruk av server side input validering for å sikre at all input er sikkert under bestemte parametere .
* Implementasjon av riktig error handling og validering kontrollere for å sikre at noen kan komme forbi sikkerhetskontrollere .
* Bruk av sikkerhetsmekanismer som som brannmurens intrusion detection system og bruke nett applikasjonens brannmurer som (WAF) ..
* Bruk av verktøy ment for sikkerhetstesting for å finne kjente sårbarheter i applikasjoner.
* Begrens bruker input data slik som rullegardinlister, avmerkingsbokser og alternativknapper i stedet for tekstfelter hvor det er mulig..
* Jevnlig oppdatering av systemet for å lukke programmets feil raskest mulig.
* [**A04:2021-Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/) is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.

A04:2021 Insecure Design fisk:

* Bruk av modellering for å identifisere potensielle sårbarheter og sikkerhetsrisikoer tidlig i designprosessen .
* Bruk av sikre designmønstre og prinsipper for sikkert design.
* Bruk referanser arkitektur som inkluderer sikkerhet best og industri standarder .
* Implementasjon av tilgangskontroll.
* Bruk av input validering og filtrering for å forhindre (XSS) cross site scripting og kommando injeksjoner.
* Bruk av sterke krypteringsalgoritmer for å beskytte sensitive data i hvile og i bruk.
* Implementasjon av riktig error handling og validering kontrollere for å sikre at noen kan komme forbi sikkerhetskontrollere.
* [**A05:2021-Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/) moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.

A05:2021 Security Misconfiguration fix:

* Vær sikker for at all programvare er satt opp og konfigurert i henhold til bransjens beste praksiser og retningslinjer.
* Ha alle programvarer som kjører på systemet oppdatert med alle de nyeste sikkerhetsoppdateringer oppdatert .
* Deaktiver og fjern alle unødvendige funksjoner ,porter eller tjenester for å gjøre slik at områdene som kan bli angrepet blir minsket.
* Bruk av sterk passord kryptering og innføre sterke passordregler.
* Lag begrensninger for tilgang til sensitive data og funksjonaliteter med bruk av tilgangskontroll og ta i bruk prinsippet for minst privilegier.
* Ta i bruk riktig error handling og logging for å oppdage og svare på sikkerhetshendelser i programmet.
* Implementering av sikkerhetsovervåkning og varsling for å se og oppdage sikkerhetshendelser. hvor man kan svare på de i nåtid.
* Innføring av sikkerhetsopplæring for å gjøre de som jobber på applikasjonen bevisst på sikkerhetsrisikoer og hvordan de kan redusere dem.
* [**A06:2021-Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/) was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

A06:2021-vulnerable and outdated components fiks:

* Gjennomfør en omfattende oversikt over alle komponentene, programvarekomponenter og biblioteker som brukes på systemet .
* Hold deg oppdatert på alle sikkerhetsoppdateringer for alle komponentene og bibliotekene som systemet tar i bruk.
* Skan regelmessig etter kjente sårbarheter på komponentene som blir tatt i bruk i systemet og prioriterer de gir høyest sikkerhetsrisiko.
* Lag regler for oppdateringer og oppdatering av komponenter så fort som mulig ved oppdagede sikkerhetsfeil.
* Ta i bruk av systemer som automatisk søker, oppdager og varsler om sikkerhetshull i komponentene .
* Lag regler for regelmessig testing og validering av eventuelle oppdateringer før de blir etablert i systemet .
* Bruk av verktøy som automatisk oppdager og fikser sikkerhetsfeil.
* Ha noen som har forståelse for viktigheten med å holde komponenter oppdatert .
* Bruk kjente og etablerte kilder for komponenter og biblioteker for pakke administratorer eller arkiv .
* Etabler en prosess for sjekk av komponenter at de er oppdaterte og fortsatt blir støttet og hvor komponentene ikke har kjente sikkerhetshull .
* [**A07:2021-Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/) was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

A07:2021 identification and authentication failures fiks:

* Implementering av multi faktor autentisering for å få mer sikkerhet.
* Bruk av sterke passord regler som krever jevnlig bytting av passord.
* Sørg for at identifiserings protokoller er oppdatert og sikre opp mot dagens standarder
* Implementering av rate limiting for å forhindre brute force angrep opp mot autentiseringsmetoden tatt i bruk.
* Bruk sikker passordlagring slik som hashing og salting.
* Jevnlig gjennomgang på bruker kontoer for å forsikre om at de som har tilgang er de eneste med tilgang til det de skal ha tilgang til .
* Etablering av protokoll om sletting av konto for brukere som ikke lenger burde ha tilgang som om noen slutter på jobben.
* Bruk av sikre protokoller for kommunikasjon slik som SSL / TLS .
* Ha regler på arbeidsplassen hvor de forklarer simple regler for sikkerhet som å ikke dele passord, ikke gå fra plassen sin, logge inn uten å logge av.
* Regelmessig testing av autentiserings mekanismen tatt i bruk for å oppdage og fikse sikkerhetsfeil.
* [**A08:2021-Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/) is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

A08:2021-software and data integrity failures fiks:

* Vær sikker på at oppdatering av software er verifisert av en troverdig kilde før installering.
* Bruk av sikre protokoller for kommunikasjon slik som SSL / TLS for å beskytte mot programvareoppdatering for kritisk data som blir overført.
* Ha tilgangskontroll for tilgang til viktig data og begrense tilgang til systemets komponenter.
* Bruk signering for å autentisere at det er en programvareoppdatering og at det ikke har blitt tuklet med.
* [**A09:2021-Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn’t well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.

A09:2021-Security Logging and Monitoring Failures fiks:

* Etablering logg og overvåkningsplan som sjekker alle systemer og applikasjoner . Her skal det være en oversikt over hvor og hva som skal bli logget.
* Bruk automatiserte verktøy og scripts for å samle inn analysere loggdata. det verktøyet burde også varsle sikkerhetsteamet om noe ikke stemmer.
* Regelmessige sikkerhetskopier av logger burde bli tatt i bruk. hvor de burde bli lagret sikkert.
* Ha en rutine for trusler for å identifisere angreps scenarier og prioritering logging av dem etter hvor alvorlig sikkerhetsnivået .
* Regelmessig testing og overvåking over lydsystemet .
* Alle sikkerhetshendelsene i systemet blir logget og sporet og å ha en plan for å håndtere situasjonen
* Tren de som jobber på prosjektet opp til å ta i bruk logging og hvordan logging kan identifisere sikkerhetstrusler.
* [**A10:2021-Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/) is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it’s not illustrated in the data at this time.

A10:2021-Server-Side Request Forgery issue fiks:

* Lær utviklere om (SSRF) server-side request forgery hva sårbarheten med det er og påvirkningen på systemet.
* Sikre slik at innkommende forespørsler er validert riktig og at de tar i bruk teknikker mot SSRF.
* Bruk av inndata-tekniker som eksempel: whitelist input data for å sikre gyldige data blir sendt inn.
* At applikasjonen skal bruke minst mulig privilegier og tillatelser for minimere skaden om vellykket SSRF-angrep skjer.
* Implementere tilgangskontroll mekanismer for å begrense forespørsler som kan gjøres fra applikasjonen.
* Overvåk systemet for SSRF forsøk og bruk av tjenester for å blokkere uønskede forespørsler.
* Bruk av sikker kodings praksis for å unngå bruk av ting slik som bruker input i system kommandoen og bruk av sikre sikkerhetsprotokoller som HTTPS.

Det skrevet på engelsk er direkte hentet fra OWAST sin side Link: https://owasp.org/www-project-top-ten/