Authentication (Digital identity)

SYSTEMATIC WAY TO SAFEGUARD OUR ASSETS

Identify

- Assets – (inventory system, asset management)
- Potential threats – (threat modeling, risk assessment)

Prevent

- Block attacks – (authentication, access control, encryption, firewall)
- Reduce vulnerabilities – (static/dynamic code analysis, software updates, vulnerability penetration testing)

Detect

- When an incident happens or shortly after – (monitoring systems, logs, malware scan, intrusion detection systems, integrity checksums/hashes, digital signatures)

Respond

- Be able to respond to stop attacks and prevent further damage – (intrusion prevention systems, shutting down and rebuilding the system)

Recover

- Correct copy of the data can be reloaded from backup

1) Identification

a. Det er en metode hvor noen (bruker, program eller prosess) hevder å ha en spesifikk identitet (navn, bruker, nummer eller e-post). Systemet vil vite hvem du er. Identifikasjon (ID) bestemmer om brukeren er autorisert tilgang til systemet eller ikke. Bestemmer bruker privilegier. Brukes i tilgang kontroll til ressurser. Hver bruker burde være unik og ikke være fysisk forklarende.

Å håndtere bruker identiteter er noe av det mest kompliserte innenfor organisasjoner. Identity management er en del av noe større som heter **Identity & Access Management (IAM).**

IAM tar vare på management med identiteter og rettigheter

2) Authentication

Autentisering er måten en bruker bekrefter sin identitet overfor systemet. Målet med et autentiseringssystem er å verifisere at brukeren faktisk er den de påstår å være. Med andre ord, bekrefte identiteten (ID).

Det er flere måter å autentisere seg på i et datamaskinsystem:

- Noe brukeren vet:
  - Passord, PIN-kode, sikkerhetsspørsmål
- Noe brukeren har:
  - Nøkkelkort, smartkort, kryptografisk nøkkel, mobiltelefon, engangspassord (OTP)-programvare eller maskinvare, kodebrikke
- Noe brukeren er eller gjør:
  - Statiske biometriske data, som fingeravtrykk, netthinne og ansiktsgjenkjenning
  - Dynamiske biometriske data, som tale/lydmønster, håndskrift, tastetrykkmønstre

- Elektronisk overvåking:
  - Angriperen fanger opp passord som sendes over nettverket. Mottiltak:
  - Krypter kommunikasjon som sender passord over internett.
- Tilgang til passordfilen:
  - Angriperen skaffer tilgang til passorddatabasen og bruker deretter en bruteforce-angrepsmetode for å finne passordene. Mottiltak:
  - Kontroller tilgangen til passorddatabasen; generer nye passord hvis de kompromitteres; bruk sterke hash-algoritmer og salter.
- Brute-force-angrep:
  - Prøver populære og svake passord med mange bruker-ID-er. Dette kalles også "Password Spraying." Mottiltak:
  - Sjekk passordvalget; blokker datamaskiner som gjør flere forsøk; implementer flerfaktorautentisering hvis mulig.
- Ordbokangrep:

    - Filene med tusenvis av ord sammenlignes med brukerens passord til en kamp er funnet. Mottiltak:
    - Sjekk passordvalget; opplæring av brukere i passordvalg.
- Regnbuetabell:
    - En angriper bruker en tabell som inneholder alle mulige passord allerede i en hash-format. Mottiltak:
    - Sørg for at salt (tilfeldig verdi) brukes. Bruk hash-algoritmer spesielt utviklet for passord.
- Datatyveri:
    - Angriperen får tilgang til datamaskinen som brukeren er logget på. Mottiltak:
    - Logg ut automatisk etter en periode med inaktivitet.
- Utnyttelse av brukerfeil:
    - Angriperen bruker sosial manipulasjonstaktikk for å villede brukeren. Mottiltak:
    - Brukeropplæring.
- Utbytting av flere passord:
    - Det samme passordet brukes på forskjellige systemer/kontoer, noe som gjør det enklere for en angriper å få tilgang til ressurser når ett passord blir kompromittert. Dette kalles også "Credential Stuffing." Mottiltak:
    - Implementer flerfaktorautentisering.

Bruk passordkontroller-/knakkere:

- Sjekk regelmessig brukerens passord og informer dem om svake passord.

Passordhashing og kryptering:

- Sørg for at passordfilen/databasen ikke lagres i klartekst. Bruk krypterings- eller hash-algoritmer.

Passordaldring:

- Angi en utløpsdato slik at passord kun er gyldige i en bestemt periode.

Begrens påloggingsforsøk:

- Begrens antall påloggingsforsøk.

Bruk datagenererte passord:

- Generer tilfeldige passord (men dette aksepteres ofte dårlig av brukere).

Brukeropplæring:

- Forsikre deg om at brukere blir informert og er klar over viktigheten av å velge passord som er vanskelige å gjette. Gi råd til brukere om strategier for passordvalg.

Bruk passordbehandler:

- En passordbehandler er et verktøy/programvare som hjelper en bruker med å kryptere, lagre og administrere passord selv. Verktøyet hjelper også brukeren med å lage vanskelige og sikre passord. Noen verktøy tilbyr muligheten til automatisk pålogging på nettsteder som krever passordautentisering.

Password checkers er verktøy som utgjør dictionary eller brute-force angrep for å finne svake passord.

PASSWORD HASHING: De fleste operativsystemet lagrer ikke passord i klar tekst. De er lagret som hash av type MD5 eller SHA for ekstra sikkerhetstiltak. Hash algoritmer vil alltid produsere samme hash for samme tekst. Derfor har vi salter som betyr at samme passord ikke kan lagres likt.

Selv om SHA-hashalgoritmene er raske og anbefales for mange formål, er de ikke egnet for hashing av passord. Hvis en angriper får tak i lagrede passordhasher, kan de gjennomføre brute force-angrep på hashene offline. CPUs, kraftige grafikkort (GPUs) og spesialtilpasset maskinvare kan beregne milliarder av hash-verdier per sekund. En bedre alternativ er å bruke hash-algoritmer som er spesielt designet for passord, som for eksempel:

- PBKDF2, som anbefales av NIST.
- Argon2id.
- scrypt.
- bcrypt.

Disse algoritmene krever mye ressurser og er utviklet med det formål å bremse offline brute force-angrep på hash-verdier.

3) Authorization

4) Resource

5) Accountability