Øvelse 54 – Design af detekteringspipeline

Detekteringens 7 abstraktionslag

Formål

Formålet med øvelsen er at træne jeres evne til at tænke systematisk om detektering af angreb – med udgangspunkt i modellen *detekteringens 7 abstraktionslag*.

Fokus er ikke at anvende modellen mekanisk, men at bruge den som et værktøj til at:

- analysere hvordan angreb manifesterer sig i systemer
- omsætte observationer til detekterbare events
- strukturere en sammenhængende detekteringspipeline

Øvelsen styrker forståelsen for:

- hvordan angreb efterlader spor
- hvordan overvågning, klassificering og alarmering hænger sammen
- hvordan man træffer beslutninger under usikkerhed i detektering

Det, I lærer her, er en måde at tænke detektering på – som kan anvendes til at analysere, designe og forklare en detekteringsstrategi.

Modellen kan anvendes som struktur i jeres projekt, men det er jeres faglige begrundelser, refleksioner og evne til at skabe sammenhæng i detekteringen, der vægtes. Modellen er et værktøj til at træne denne tankegang – ikke et facit.

Tidsramme

Ca. 60 minutter

Information

Øvelsen er individuel. Opsamling laves i grupper i næste øvelse – se tidsplanen.

Denne øvelse er en grundlæggende del af detection engineering – den faglige disciplin, der handler om at udvikle, justere og dokumentere systematisk detektering af angreb og trusler.

Øvelsen introducerer og anvender modellen detekteringens 7 abstraktionslag som metode til:

- analyse af angreb
- design af detekteringspipelines
- dokumentation af detekteringsstrategier

Modellen kan anvendes både som analyseværktøj, dokumentationsstruktur og designmetode i arbejdet med detekteringsregler, logning og hændelseshåndtering.

 Note om modellen

Modellen for detekteringens 7 abstraktionslag er baseret på teori fra bogen *Engineering trustworthy systems: Get cybersecurity design right the first time*(Saydjari, 2021, kap. 13), men er i denne øvelse anvendt i en pædagogisk tilpasset form.

Formålet med tilpasningen er at gøre modellen mere anvendelig som:

- analyseværktøj
- designmetode
- dokumentationsstruktur

I praksis vil grænserne mellem lagene ofte være mindre tydelige, og flere lag kan overlappe.

 Fokus i denne øvelse er derfor ikke en "perfekt opdeling", men en fagligt begrundet og sammenhængende detekteringspipeline.

 Begreb: Event vs. hændelse

I denne øvelse anvendes begrebet **event**, som også bruges i den faglige litteratur og i praksis (fx i logs og SIEM-systemer).

Et *event* kan forstås som en **hændelse i systemet**, der kan observeres i data.

Begreberne dækker over det samme, men vi bruger **event** for at være konsistente med den terminologi, der anvendes i detection engineering.

Grundlæggende begreber i detektering

For at arbejde systematisk med detektering er det vigtigt at forstå, hvordan evidens for angreb opstår i praksis.

I de fleste tilfælde er enkeltstående events ikke nok til at afgøre, om noget er et angreb. Detection handler derfor om at arbejde med **korrelation** og **signalstyrke**.

Korrelation (sammenhæng mellem events)

Korrelation betyder, at flere events ses i sammenhæng og tilsammen giver stærkere evidens for et angreb.

Et enkelt event kan ofte forklares som normal adfærd. Når flere relaterede events optræder sammen – fx over tid, fra samme bruger eller samme system – bliver det mere sandsynligt, at der er tale om et angreb.

Eksempel:

- Flere fejlede autentificeringsforsøg inden for kort tid
- Efterfulgt af en succesfuld autentificering fra samme kilde
- Efterfulgt af brug af superbrugerrettigheder

Hver event kan være legitim.
I kombination kan de indikere kompromittering.

Det afgørende er, at events **ikke kun optræder samtidig**, men at de **gør hinanden mere mistænkelige**.

Signalstyrke (stærke og svage signaler)

Når angreb manifesterer sig i et system, varierer tydeligheden af de spor, de efterlader.

- **Stærke signaler** er tydelige indikatorer på angreb
  (fx kendte malware-signaturer eller netværksforbindelse til en kendt trusselsaktørs server)

Tommelfingerregel:
Hvis signalet kan være normalt, er det ikke et stærkt signal

- **Svage signaler** er tvetydige og kan både være legitime og ondsindede
  (fx autentificeringsforsøg, netværksforbindelser eller brug af administrative rettigheder)

Svage signaler bliver typisk først stærke, når de ses i sammenhæng med andre signaler.

I praksis består detektering ofte af flere svage signaler, som først bliver betydningsfulde, når de kombineres.

Vigtig pointe: Korrelation, signalstyrke og flere pipelines

I praksis hænger **korrelation** og **signalstyrke** tæt sammen med, hvordan detektering er designet.

Svage signaler opstår ofte i **forskellige dele af systemet** og dermed i **forskellige datakilder og pipelines**.

Eksempel:

- Én pipeline identificerer fejlede autentificeringsforsøg
- En anden pipeline identificerer succesfuld autentificering fra samme kilde
- En tredje pipeline identificerer privilegeret aktivitet

Hver pipeline producerer typisk **svage signaler**, som isoleret set kan være legitime.

 Det er først, når disse **kombineres (korreleres)**, at signalstyrken øges, og der opstår reel evidens for et angreb.

 Det betyder:

- detektering består ofte af **flere del-pipelines**, ikke én samlet
- Hver pipeline bidrager med **delvis evidens**
- Det er i **attack detection-laget**, at disse samles og vurderes

 Derfor er det vigtigt ikke kun at tænke i enkelte events eller regler, men i **hvordan flere svage signaler fra forskellige kilder kan forstærke hinanden**

Central pointe

> Ét event er sjældent nok.
> Det er kombinationen af flere signaler, der skaber evidens for et angreb.

Disse begreber er centrale i arbejdet med øvelsen og bør indgå i dine overvejelser, når du designer din detekteringspipeline.

Hvad er en detekteringspipeline?

En detekteringspipeline er den kæde af trin, hvor system- og netværksdata omsættes til en meningsfuld alarm:

Data → Events → Detection (angrebsvurdering) → Alarm → Reaktion

Formålet er at opdage angreb hurtigt og præcist – og samtidig filtrere støj for at undgå falske positiver.

Detekteringens 7 abstraktionslag

Lag 1: Feature Selection
Spørgsmål: Hvor og hvordan manifesterer angrebet sig? Hvilken adfærd observeres der?

Lag 2: Feature Extraction
Spørgsmål: Hvor findes informationen?

Lag 3: Event Selection
Spørgsmål: Hvilken del af datastrømmen er relevant?

Lag 4: Event Detection
Spørgsmål: Hvad definerer en konkret event?

Lag 5: Attack Detection
Spørgsmål: Hvornår er det et angreb?

Hvilke events kombineres, hvor mange, og over hvilken tidsperiode?

Lag 6: Attack Classification
Spørgsmål: Hvilken type angreb er det?

Lag 7: Attack Alarming
Spørgsmål: Hvem alarmeres og hvordan?

Eksempel 1: Mistænkelige autentificeringsforsøg

For at forstå sammenhængen mellem lagene, ses her et simpelt eksempel:

Lag	**Eksempel (SSH brute force)**
Feature Selection	Mange fejlede autentificeringsforsøg (svagt signal)
Feature Extraction	auth.log
Event Selection	fejlede autentificeringsforsøg
Event Detection	Gentagne fejlede autentificeringsforsøg fra samme IP over kort tid
Attack Detection	Mulig brute force baseret på mange fejlede autentificeringsforsøg fra samme kilde over tid

Lag	Eksempel (SSH brute force)
Attack Classification	T1110
Attack Alarming	Alarm ved fortsat aktivitet eller efter succesfuld autentificering

 Eksemplet illustrerer, hvordan flere svage signaler (fejlede autentificeringsforsøg) over tid kan kombineres til evidens for et angreb.

Eksempel 2: Mistænkelig brug af privilegeret adgang

Lag	Eksempel
Feature Selection	succesfuld autentificering og privilegiebrug (svage signaler)
Feature Extraction	auth.log, sudo logs
Event Selection	succesfuld autentificering, sudo
Event Detection	succesfuld autentificering efterfulgt af brug af superbrugerrettigheder
Attack Detection	Mistænksomme kommandoer og privilegeret handlinger
Attack Classification	Privilege escalation
Attack Alarming	Alarm ved mistænkelig privilegeret aktivitet

 Eksemplet her viser meget svage signaler, som kan være en del af normal brug.

 Disse to eksempler kan korreleres for at skabe stærkere evidens.

Korrelation mellem eksemplerne

Hvis de to mønstre optræder i sammenhæng:

- Gentagne fejlede autentificeringsforsøg

- Efterfulgt af succesfuldt autentificeringsforsøg
- Efterfulgt af privilegeret aktivitet

 bliver evidensen markant stærkere.

 Det er ikke de enkelte events, men **kombinationen og rækkefølgen**, der indikerer et muligt angreb.

Instruktioner

Øvelsen udføres individuelt.
I den næste øvelse præsenteres og diskuteres resultatet i grupper.

 Krav til besvarelse:

- Hvert lag skal bygge videre på det foregående
- Undgå generelle svar – vær konkret
- Tænk i sammenhæng – ikke isolerede svar
- Du skal inkludere mindst ét eksempel på korrelation mellem events og forklare hvorfor kombinationen er stærkere end hvert event alene

---

 Før du går i gang – sådan tænker du detektering

Før du begynder at udfylde modellen, er det vigtigt at forstå én ting:

 Detektering starter ikke med logs – det starter med angrebets adfærd.

Det centrale spørgsmål er:

“Hvis jeg var trusselaktør – hvad ville jeg gøre, og hvilke spor ville det efterlade i systemet?”

Tænk angrebet igennem som en sammenhængende proces:

Formål: Hvad forsøger angriberen at opnå? Handlinger: Hvad gør angriberen konkret? Systempåvirkning: Hvad ændrer sig i systemet? Observation: Hvor kan disse ændringer ses?

 Først derefter giver det mening at tale om logs og data.

Trin 1 – Vælg et overordnet mål for trusselaktøren

Udvælg et mål, som din trusselaktør ønsker at opnå.

Eksempler:

- Opnå vedvarende adgang til et system
- Eskalere privilegier til root/administrator
- Eksfiltrere følsomme data
- Manipulere systemkonfiguration

Trin 2 – Skabelon

Du skal arbejde systematisk gennem alle 7 lag og beskrive, hvordan angrebet:

- manifesterer sig i systemet
- kan observeres og udvælges i data
- kan detekteres og vurderes
- kan klassificeres og alarmeres

Du arbejder med de 7 lag fra tidligere i øvelsen. Fokus er ikke at gengive modellen, men at anvende den til at skabe en sammenhængende detekteringspipeline.

Arbejd nedefra og op (fra 1 → 7), og sørg for at hvert lag bygger videre på det foregående.

Der er ikke ét korrekt svar – fokus er på din faglige begrundelse og sammenhængen i din pipeline.

Refleksionsspørgsmål

- Hvilket lag var mest uklart at definere – og hvorfor?
- Giv ét konkret eksempel på en falsk positiv i jeres pipeline
- Hvordan kunne en angriber forsøge at undgå jeres detektion?
- Hvor i din pipeline er du “blind”?
- Hvordan kunne jeres pipeline forbedres?
- Hvordan kan I bruge denne tilgang i eksamensprojektet?

Afsluttende opgave

Skriv kort (5–10 linjer):

Forklar din pipeline som en samlet detekteringsstrategi

- Hvordan hænger lagene sammen?
- Hvor er styrker og svagheder?

Referencer

Saydjari, O. S. (2021). *Engineering trustworthy systems: Get cybersecurity design right the first time*. McGraw-Hill Education.

Last update: 2026-04-12 17:32:46