Machine learning techniques for flow-based intrusion detection systems

...

Axel Faes: Bachelorthesis

Doelstelling ID systemen

Classificeren/detectie van onverwacht netwerkgedrag

Extern:

- port scans
- ssh connection attempts
- side-effect verkeer (ICMP, IRC)
- high volume DDoS
- low volume DDoS

Intern:

- botnets (communication with master/assist with DDoS attacks)
- worms (bij binnendringen/uitbreken van systeem)

Typische werking van bestaande ID systemen

Veel voorkomende technieken:

- Signature-based detecties (op basis van rule matching op inhoud van flow/packet data)
- Anomaly detection

Algemene doelstelling : patroonherkenning

- Binaire classificatie: malicious vs non-malicious
- Classificatie voor specifieke type van malicious behaviour

Specifieke eigenschappen v.h. te ontwikkelen systeem

Detectie louter gebaseerd op flow data:

- i.t.t packet- en log-based ID systemen
- specifiek bedoeld voor high traffic systems
- vereist geen in-depth knowledge van het netwerk

Gebruik van machine learning technieken

- Kosten-efficiënt inzetten in bestaande netwerken
- Algoritmes ‘leren’ zelf zonder regeltjes expliciet te programmeren

De onderzoeksvragen van de bachelorproef

- In hoeverre zijn machine learning technieken inzetbaar voor anomaly detection (welke technieken werken goed/niet goed) ?
- Hoe kan flow data gebruikt worden in deze technieken?
- Kunnen we een IDS maken dat out-of-the-box een aanvaardbare ‘hit rate’ biedt ?
- Welke types anomalie kunnen we automatisch detecteren ?
- Is (automatische) klassificatie van de anomalie mogelijk ?
- Zijn dergelijke technieken bruikbaar in real-life condities ?

Werking:

Stap 1

Aanleren van het model via subset van een learning data set

Stap 2

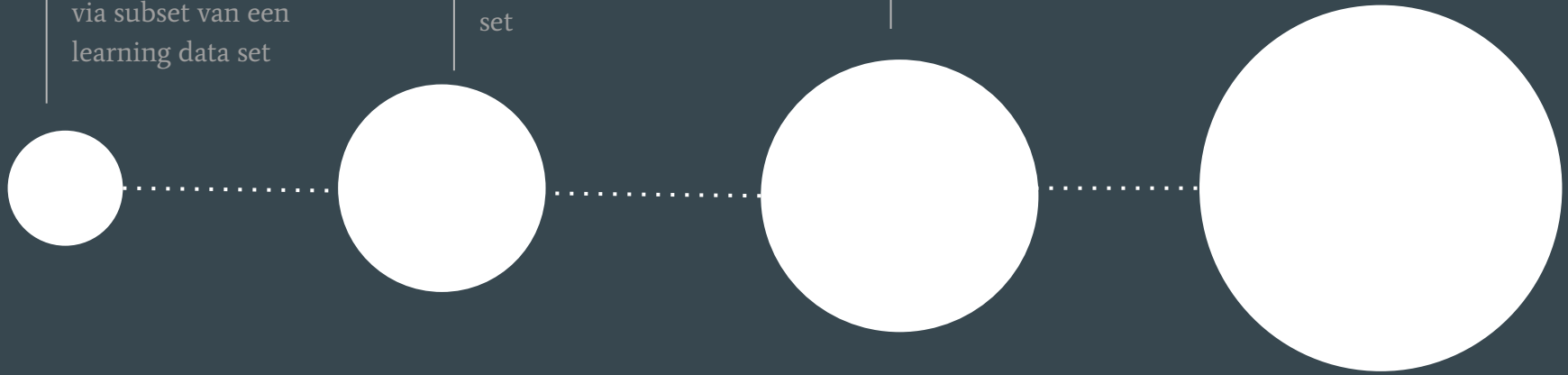
Validatie met gekende test-data uit learning data set

Stap 3

Testing met real-world gelabelde data

Stap 4

Validatie met ongelabelde real-world data



Training data

Geannoteerde data sets zijn specifiek bedoeld om het algoritme een model aan te leren (stap 1 en 2)

- worden typisch opgedeeld in disjuncte subsets
- 1 subset specifiek om een model aan te leren
- 1 subset om dit model te kunnen valideren

Momenteel gebruikte datasets

CTU-13 dataset (stappen 1, 2 en 3):

- Bevat botnet, normaal en background traffic
- Zeer gedetailleerd geclassificeerd

Tracelabel dataset van UTwente (stappen 1, 2 en 3):

- Bevat traffic geclassificeerd als malicious door honeypot
- Bevat ftp, http, ssh, icmp, irc verkeer

EDM dataverkeer (stap 4):

- Unlabeled data
- Manuele verificatie van classificatie

Vragen (1)

- Welke classificatie van ‘onverwachte traffiek’ gebeurt er momenteel in datacenter/hosting context ?
- Welke informatie is interessant om te identificeren en welke niet ?
- Hoe gebeurt classificatie ?
  - manueel vs automated (en dmv welke tools/systemen ?)
  - gebaseerd op welk type informatie (flow vs logs vs ...)
- Welke bijkomende (m.a.w. nu niet aanwezige) classificatie zou interessant zijn in datacenter context ?
- False positives vs negatives, wat is voor de Cegeka context het minst gewenst ?
- Welke ‘hit rate’ is wenselijk/aanvaardbaar voor een dergelijk systeem ?

Vragen (2)

- Welke data kan ev. beschikbaar gesteld worden ? noot: typisch zijn er 2 types nodig:
  - geannoteerde datasets (voor stap 3)
  - niet-geannoteerde control datasets (voor stap 4) : manueel verifieerbaar
- Is het mogelijk om output van een bestaand IDS te verkrijgen (bij voorkeur gekoppeld aan geannoteerde/niet geannoteerde datasets die hierboven vermeld zijn) ?
  - belangrijke stap voor quantificering van de efficiëntie van het systeem tov een ‘real world’ IDS (bv gebaseerd op signatures)
- Is in tweede instantie een manuele verificatie mogelijk (i.c. voor kleine subset) ?