Network Anomaly Detection

Anomaly Detection Challenges WS'16/17

Mohammad Reza Norouzian, Bojan Kolosnjaji, George Webster
Chair of IT Security (I20)
Department of Informatics
Technische Universität München

November 29, 2016

Introduction

- We are drowning in the deluge of data that are being collected world-wide, while starving for knowledge at the same time
- Anomalous events occur relatively infrequently
- However, when they do occur, their consequences can be quite dramatic and quite often in a negative sense

“Mining needle in a haystack. So much hay and so little time”

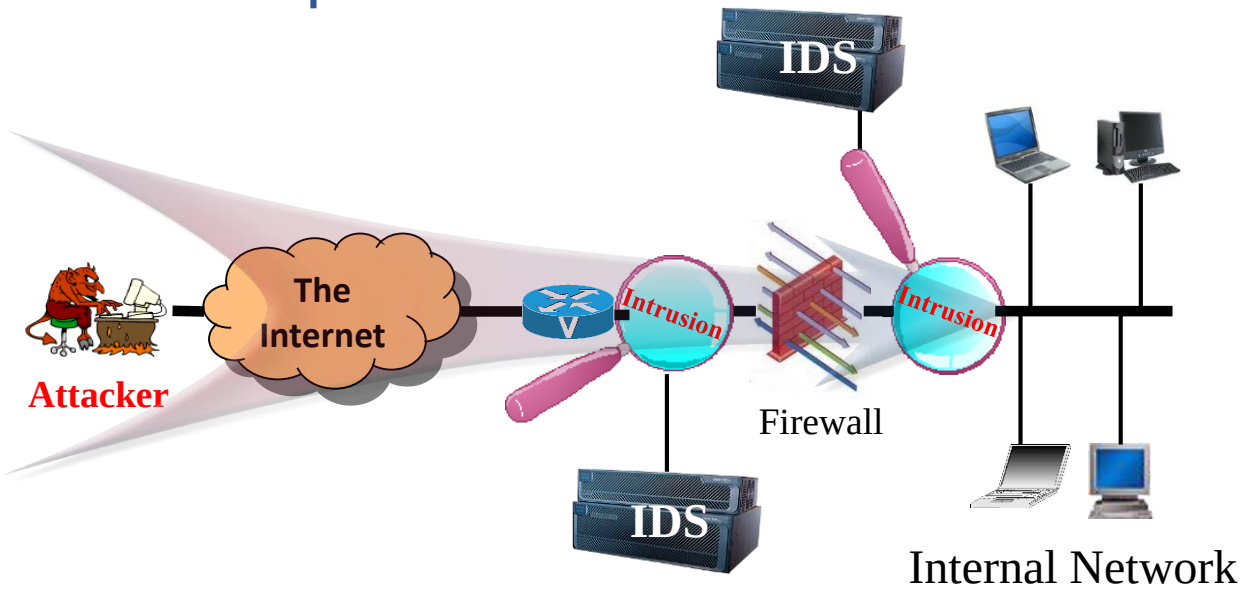
What's an Intrusion?

- Successful attack is usually (but not always) associated with an access control violation
  - A buffer overflow has been exploited, and now attack code is being executed inside a legitimate program
  - Outsider gained access to a protected resource
  - A program or file has been modified
  - System is not behaving "as it should"

- The goal of an intrusion detection system (IDS) is to detect that bad things are happening (intrusion)
  - Just as they start happening (hope so)
  - How is this different from a firewall?

Intrusion detection styles

- Misuse detection: precise descriptions of known malicious behavior.
- Anomaly detection: have a notion of normal activity and flag deviations from that profile.



- *Specification-based detection: defining allowed types of activity in order to flag any other activity as forbidden.

Detection Styles in Actual Deployments

- Striking imbalance deployments:
  - Almost exclusively only misuse detectors in use
  - Detect signatures (characteristic byte sequences)

- Question:
  - However, anomaly detection is extremely appealing (in the literatures)
    - Promises to find novel attacks w/o anticipating specifics
    - Machine learning works so well in other domains
  - But it's hard to find any machine learning NIDS in real-world deployments, why?

Misuse Detection (Signature-Based)

- Set of rules defining a behavioral signature likely to be associated with attack of a certain type
  - Example: buffer overflow
    - A setuid program spawns a shell with certain arguments
    - A network packet has lots of NOPs in it
    - Very long argument to a string function
  - Example: SYN flooding (denial of service)
    - Large number of SYN packets without ACKs coming back
    - …or is this simply a poor network connection?
- Attack signatures are usually very specific and may miss variants of known attacks
  - Why not make signatures more general?

Anomaly Detection

- Originally introduced by Dorothy Denning in 1987
  - Assumption: attacks exhibit characteristics NOT observed for normal usage
  - Propose: host-based IDS
    - Host-level system building per-user profiles of activity
    - E.g., login frequency, session duration, resource consumption
- Machine learning (ML):
  - Training: trained with reference input to “learn” its specifics
    - Supervised or unsupervised
  - Test: deployed on previously unseen input for the actual detection process

Anomaly Detection Cont'd

- Define a profile describing "normal" behavior
  - Works best for "small", well-defined systems single program rather than huge multi-user OS
- Profile may be statistical
  - Build it manually (this is hard)
  - Use machine learning and data mining techniques
    - Log system activities for a while, then "train" IDS to recognize normal and abnormal patterns
  - Risk: attacker trains IDS to accept his activity as normal
    - Daily low-volume port scan may train IDS to accept port scans
- IDS flags deviations from the "normal" profile

Machine Learning in Other Domains

- Examples (for comparison):
  - Amazon/Netflix – product recommendation
  - OCR (optical character recognition) systems
  - Natural language translation
  - Spam detection

- Claim: the task of finding attacks is fundamentally different from other applications
  - Making it significantly harder for us to employ ML effectively

Machine Learning in Intrusion Detection

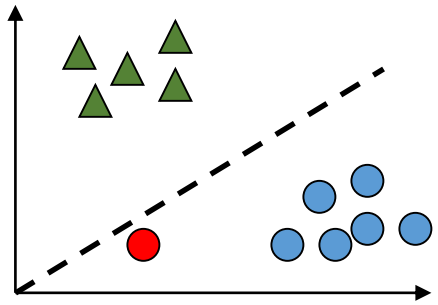
- Some well-known problems:
  - High false positive rate
  - Lack of (attack-free) training data
  - Attackers can try to evade detection
- Goal:
  - Using anomaly detection effectively in the real world operational environments (for network intrusion detection)

Challenges of Using Machine Learning

- Outlier Detection
- High Cost of Errors
- Semantic Gap (interpretation of results)
- Diversity of Network Traffic
- Difficulties with Evaluation
  - Training Data
  - Semantic Gap
  - Evasion Risk

C1. Outlier Detection

- ML is good at finding similarities (a classification problem)
  - E.g., Amazon – recommend similar products
    - Similarity: products that tend be brought together
  - However, anomaly detection requires to discover meaningful outliers
- Outlier detection is also a classification problem
  - → Normal vs. not normal
  - But, instances of ALL classes in the training data are required
    - e.g., ML-based spam detection is more successful



C2. High Cost of Errors

- FP → taking more efforts in determining that the report reflects benign underlying activity
- FN → potential to cause serious damage to an organization
- In contrast:
  - (Ex. 1) Product recommendation systems
    - FP → deliberately promote other products
    - FN → continue shopping, rather than switching to different seller
  - (Ex. 2) OCR technology
    - Statistical language models allows for post-processing
    - In addition users have been trained not to expect too much
  - (Ex. 3) Spam detection
    - FP can be very expensive, but FN do not have a significant impact

C3. Semantic Gap

- How to transfer results into actionable reports for the network operator?
- Q: abnormal activity (FP) or attack (TP)?
  - By definition, a ML Algo. does NOT make any mistakes within its model of normality (seen before);
  - yet for the operator it is the results' interpretation that matters
- Local security policies:
  - Many security constraints are a site-specific property
  - Other technical details (threshold selection)
- → Need to understand how the features relate to the semantics of the network environment

C4. Diversity of Network Traffic

- Even within a single network, the network's most basic characteristics can express massive variability
  - For an anomaly detection system, such variability make it difficult to find a stable notion of "normality"
- Reduce diversity → data aggregation !!
  - Anomaly detection systems operate on highly aggregated information (e.g., volume per hour)
  - → non-ML approaches might work equally well
- Diversity is not restricted to packet-level features, but extends to application-layer information as well

C5. Difficulties with Evaluation

- Designing sound evaluation schemes is not easy
  - 1) Finding the Right Data
    - Two old and uninteresting dataset: DARPA and KDD Cup
    - Why is difficult to find the right dataset?
      - Privacy concern!
    - Simulation, anonymization, data collection in a small network
  - 2) Mind the Gap
  - 3) Adversarial Setting
    - Concerning the adversarial environment the NIDS operate in
    - Attackers adjusting their activity to avoid detection (especially cheating the ML algo.)

C6. Lack of Training Data

- Attack free data hard to obtain
- Labeled data expensive to obtain
- Synthetic Vs. Real life traffic
- Nature of attributes
  - Binary
  - Categorical
  - Continuous
  - Hybrid

	categorical	continuous	categorical	continuous	binary
Tid	**SrcIP**	**Duration**	**Dest IP**	**Number of bytes**	**Internal**
1	206.163.37.81	0.10	160.94.179.208	150	**No**
2	206.163.37.99	0.27	160.94.179.235	208	**No**
3	160.94.123.45	1.23	160.94.179.221	195	**Yes**
4	206.163.37.37	112.03	160.94.179.253	199	**No**
5	206.163.37.41	0.32	160.94.179.244	181	**No**

Data Collection

- Type of features: Source and destination IP addresses, ports, packet headers, network traffic statistics Tools
- Tools:
  - Tcpdump: command line tool
  - Bro IDS: open source based network monitoring framework.
  - Snort: open source IDS packet capture and signature matching
  - Wireshark: popular open source packet sniffer Data Collection

No.	Time	Source	Destination	Protocol	Info
199	63.957563	72.14.207.104	192.168.0.81	TCP	www > 51512 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=.
200	63.957622	192.168.0.81	72.14.207.104	TCP	51512 > www [ACK] Seq=1 Ack=1 Win=5840 Len=0
201	63.957761	192.168.0.81	72.14.207.104	HTTP	GET /__utm.gif?utmwv=1&utmn=894872088&utmcs=ISO-8859-1(
202	64.087354	72.14.207.104	192.168.0.81	TCP	www > 51512 [ACK] Seq=1 Ack=879 Win=7248 Len=0
203	64.091976	72.14.207.104	192.168.0.81	HTTP	HTTP/1.1 200 OK (GIF89a)
204	64.092003	192.168.0.81	72.14.207.104	TCP	51512 > www [ACK] Seq=879 Ack=330 Win=6432 Len=0
205	64.724164	192.168.0.81	84.16.81.23	ICMP	Echo (ping) request
206	64.820725	Cisco-Li_66:f3:72	Broadcast	ARP	Who has 192.168.0.75? Tell 192.168.0.1
207	65.742373	Cisco-Li_66:f3:72	Broadcast	ARP	Who has 192.168.0.75? Tell 192.168.0.1
208	65.792260	192.168.0.81	84.16.81.23	ICMP	Echo (ping) request
209	66.792294	192.168.0.81	84.16.81.23	ICMP	Echo (ping) request
210	66.814953	84.16.64.33	192.168.0.81	ICMP	Destination unreachable (Communication administrativel

Recommendations for Using Machine Learning

- In one sentence – *Understand what the system is doing!*

- Recommendations:
  - Understanding the Threat Model
  - Keep the Scope Narrow
  - Reducing the Costs
  - Evaluation
    - Working with data
    - Understanding results

R1. Understanding the Threat Model

- The threat model establishes the framework for choosing trade-offs
- Questions to address:
  - What kind of environments does the system target?
  - What do missed attacks (FN) cost?
  - What skills and resources will attackers have?
  - What concern does evasion pose?

R2. Keep the Scope Narrow

- The more narrowly one can define the target activity, the better one can make a detector to its specifics and reduce the potential for misclassifications
  - Note that ML (anomaly detection) is not a “silver bullet”

- Selecting an appropriate ML algorithm
  - Considering why the particular choice promises to perform well in the intended setting
    - Any strictly mathematical grounds?
    - Any domain-specific properties? (identifying the feature set)

R3. Reducing the Costs

- The high cost associated with each FP error often conflicts with effective operation
  - Solution: reducing the system's scope
- Requires a strategy to deal with the natural diversity of network traffic
  - Aggregating or averaging features over suitable time intervals
  - Carefully examine the features for their particular properties
- Post-process the FP (with the support of additional information)
  - e.g., BotHunter

R4. Evaluation

- 1. Working with data
  - Obtaining access to a dataset containing real network traffic from as large an environment as possible
    - Multiple of these from different networks
  - Where is the data source?
    - Recording mechanism/format/granularity…
  - Once acquired, the dataset require a careful assessment of their characteristics

R4. Evaluation (cont'd)

- 2. Understanding results
  - Manually examine FP, relate such FP to the semantics of the traffic
  - FN is harder to investigate because they require reliable ground-truth
    - Ground-truth is hard to obtain, and need to obtain at the beginning of a study
    - Use different (orthogonal) mechanism to label the input, or manually labeling
    - Final compromise: manually inject attacks
  - Also inspect the TP and TN
  - Compares results with other systems found in the literature

Conclusion

- Examines the imbalance between the study of the ML-based anomaly detection in academia, versus the lack of operational deployments of such systems
- ML for intrusion detection is challenging
  - Reasonable and possible, but need care
  - Consider fundamental differences to other domains
  - There is some good anomaly detection work out there
- Provide a set of guidelines for applying ML to network intrusion detection

Challenge 3

- Packet based Intrusion Detection
- 47 features + 2 labeled features
- 9 different category of attacks
  - Fuzzers, Analysis, Backdoors, Dos, Exploits, Generic, Reconnaissance, Shellcode and Worms
- Binary Classification of traces; normal or attack

Homework

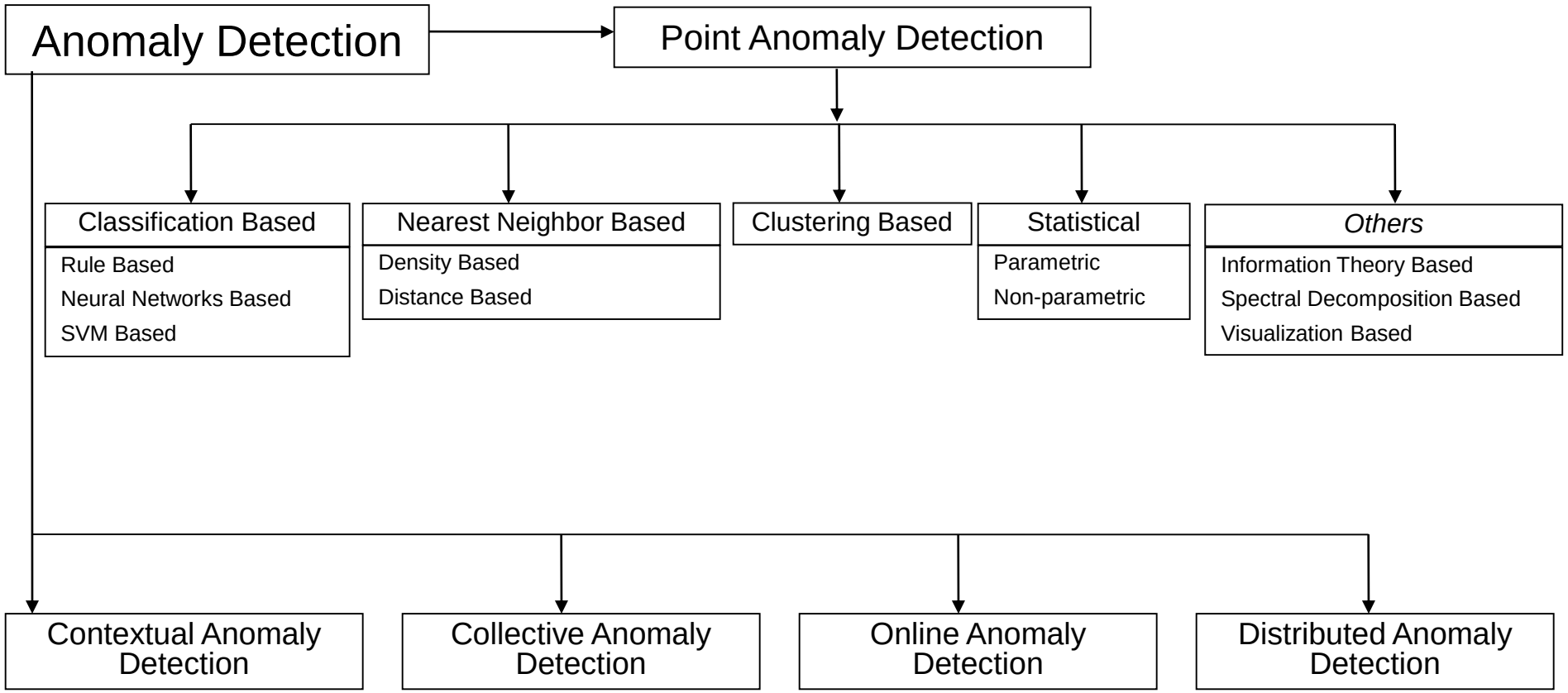
- Look at the references and answer the following questions:
- What are the challenges for network anomaly detection in compare to other domain? Which considerations/techniques as preprocessing and post processing NIDSs might be more applicable/useful?
- In NIDS domain till now most of NIDSs are using signature based techniques, what is your opinion using ML to address this issue?
- Write short answer and send it in a PDF file to my email.

References

1. Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, *0*(May), 305–316. http://doi.org/10.1109/SP.2010.25
2. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. *Communications Surveys & Tutorials, IEEE*, *16*(1), 303–336. http://doi.org/10.1109/SURV.2013.052213.00046
3. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(September), 1–58. http://doi.org/10.1145/1541880.1541882
4. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, (Cisda), 1–6. http://doi.org/10.1109/CISDA.2009.5356528
5. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Ferná Ndez, G., & Vá Zquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, *28*, 18–28. http://doi.org/10.1016/j.cose.2008.08.003

Thank you!

Taxonomy*


Anomaly Detection
Point Anomaly Detection
Classification Based
Rule Based
Neural Networks Based
SVM Based
Nearest Neighbor Based
Density Based
Distance Based
Clustering Based
Statistical
Parametric
Non-parametric
Others
Information Theory Based
Spectral Decomposition Based
Visualization Based
Contextual Anomaly Detection
Collective Anomaly Detection
Online Anomaly Detection
Distributed Anomaly Detection