Casting out Demons: Sanitising Training Data for Anomaly Sensors

Gabriela F. Cretu, Columbia University
Angelos Stavrou, George Mason University
Michael E. Locasto, Dartmouth College
Salvatore J. Stolfo, Columbia University
Angelos D. Keromytis, Columbia University

Abstract

- Obtaining clean datasets to train AD sensors has always been a problem
- The proposed technique is to include a 'sanitising' phase (does not affect the underlying AD algorithm) in the training phase of the AD sensor.
- The sanitising phase consists of creating "micro models" trained on small slices of data.

Abstract

- The micro-models are combined in a voting scheme.
- The paper shows that the sanitising phase significantly improves the quality of unlabeled data.

Introduction

- Effective AD systems require highly accurate modelling of normal data.
- Datasets are large, contain unpredictable spread of attacks, rare data and errors.
- The paper proposes a Sanitising phase, a distributed architecture for cross sanitisation, a shadow sensor for the false positive problem.

Local Sanitisation

- Feasibility of supervised and semi-supervised training?
- Unsupervised learning? Will it help to use this method?
- Remove all attacks, abnormalities and rare traffic artefacts is the first important step.

Assumptions

- Frequency of attacks is generally low relative to legitimate input
- Common attack packets tend to cluster together and form a sparse representation over time.
- Large datasets for training – increases the probability of mal-code presence.

Micro-models

- Micro-models are used in an ensemble arrangement.
- $T = \{md1, md2, \ldots, mdN\}$
- *mdi is the micro-dataset starting at time* $(i - 1) * g$ and, *g is the granularity*
- *AD:* $M = AD(T)$ *where AD can be* any chosen anomaly detection algorithm
- *micro-model,* $Mi = AD(mdi)$

Sanitised and Abnormal Models

- *Lj,i = TEST(Pj,Mi) where Pj is a packet j, Mi is the micro-model used for testing.*
- *Lj,i, has a value of 0 if the model Mi deems* the packet *Pj normal, or 1 if Mi deems it abnormal.*
- *SCORE(Pj) is the weighted score of each packet*
- split our data into two disjoint sets: one that contains only majority-voted normal packets, *Tsan and the other Tabn*

Evaluation of Sanitisation

- Measure increase in the detection accuracy of any content-based AD system when we apply training data sanitisation.
- measure the performance of the sensor with and without sanitisation.
- test each packet and consider the computational costs involved in diverting each alert to a host-based shadow sensor.

Experimental Results

Sensor	www1		www		lists	
	FP(%)	TP(%)	FP(%)	TP(%)	FP(%)	TP(%)
A	0.07	0	0.01	0	0.04	0
A-S	0.04	20.20	0.29	17.14	0.05	18.51
A-SAN	**0.10**	**100**	**0.34**	**100**	**0.10**	**100**
P	0.84	0	6.02	40	64.14	64.19
P-SAN	**6.64**	**76.76**	**10.43**	**61**	**2.40**	**86.54**

FP: false positive rate; TP: true positive rate

Sensor	www1	www	lists
A	0	0	0
A-S	505	59.10	370.2
A-SAN	**1000**	**294.11**	**1000**
P	0	6.64	1.00
P-SAN	**11.56**	**5.84**	**36.05**

signal-to-noise ratio (TP/FP); higher values mean better results

Sanitisation parameters

- The optimal operating point for any sensor can be identified automatically with offline tuning that requires no manual intervention.
- Fine tune the following: granularity of the micro-models, the voting algorithm, and the voting threshold.

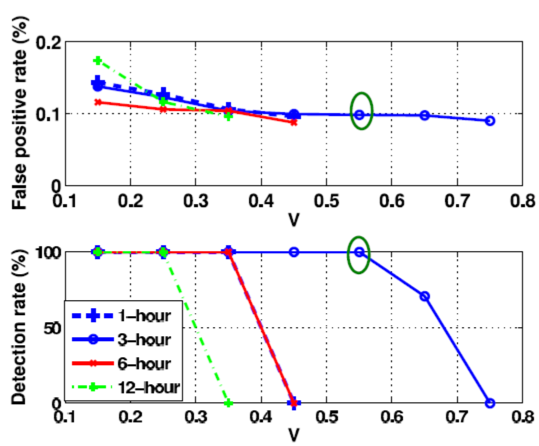

Figure 1. Performance for *www1* **for 3-hour granularity when using simple voting and Anagram** (V is the voting threshold; see section 2)

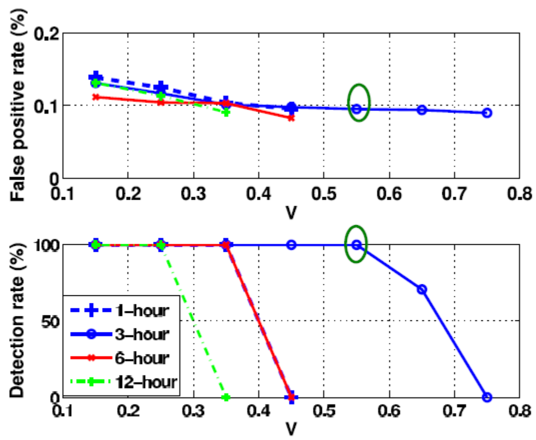

Figure 2. Performance for *www1* **when using weighted voting and Anagram** (V is the voting threshold)

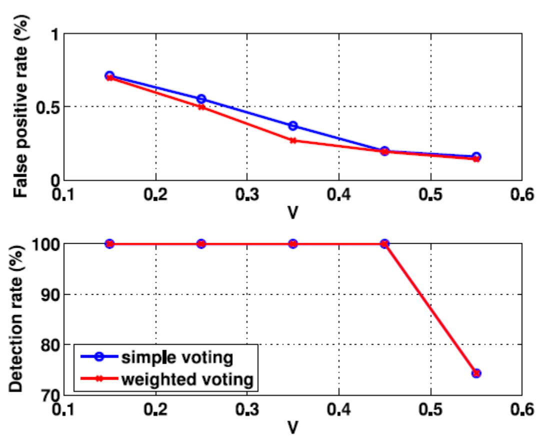

Figure 3. Performance for *www* **for 3-hour granularity when using Anagram** (V is the voting threshold)

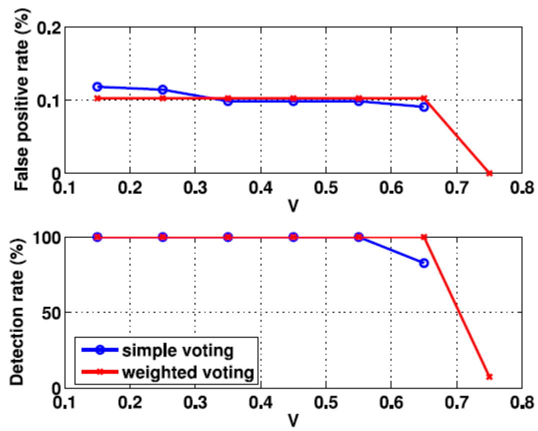

Figure 4. Performance for *lists* for 3-hour granularity when using Anagram (V is the voting threshold)

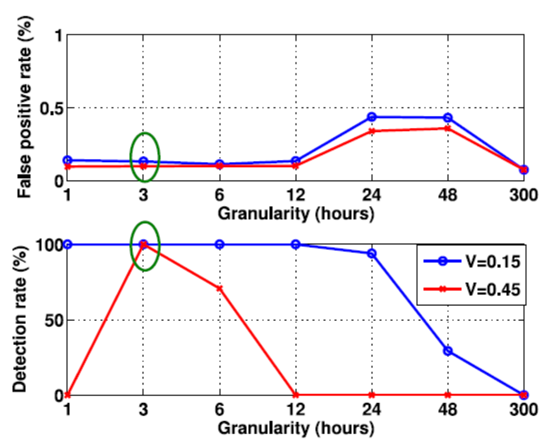

Figure 5. Granularity impact on the performance of the system for *www1* when using Anagram

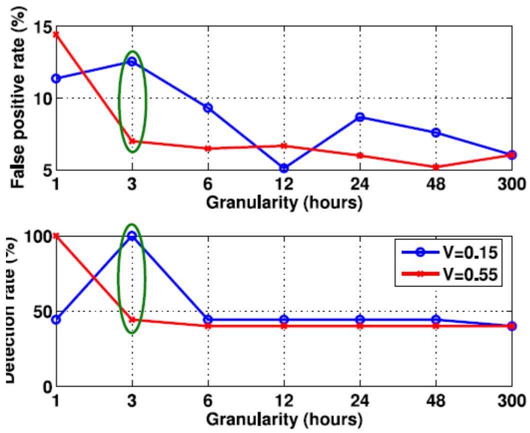

Figure 6. Granularity impact on the performance of the system for *www* when using Payl

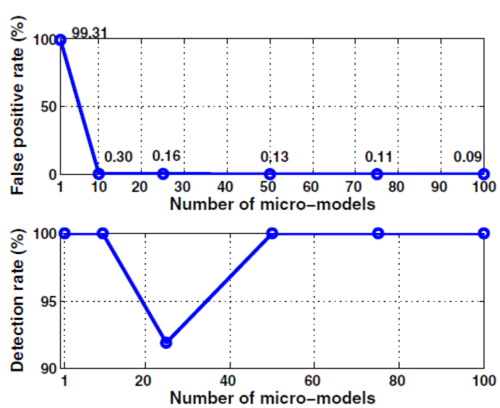

Figure 7. Impact of the size of the training dataset for *www1*

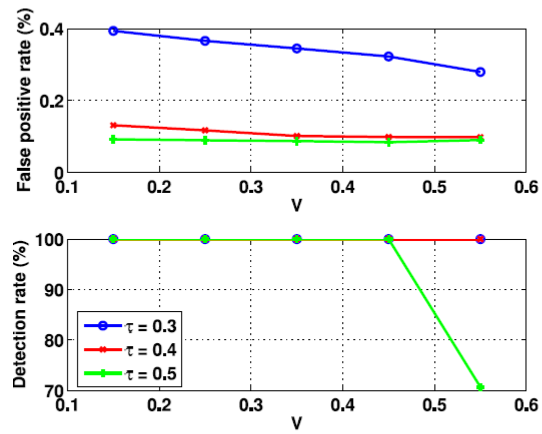

Figure 8. Impact of the anomaly detector's internal threshold for *www1* when using Anagram

Computational Performance

- Goal: keep request latency at a reasonable level, scalability
- Is the shadow sensor sufficient?
- Shadow sensor: performance, requires synchronisation of state between it and the shadowed production application and its not perfect.
- Alert rate for both Anagram and Payl does not increase by much after sanitising.

Collaborative Sanitisation

- Long-lasting attacks
- Such attacks require significant resources – effectively limits the scope of attack to a few target hosts or networks.
- Distributed system: abnormal traffic models are shared between collaborative sites.
- Cross-sanitisation improves ability to remove long living attacks.

Cross sanitised model

- Direct model differencing
- Indirect model differencing

Table 4. Recalculating sanitized and abnormal models. These routines use the abnormal models of collaborating peers to regenerate models of both normal and abnormal local data.

ROUTINE CROSSSANITIZED() $\forall i \in [1..M]$ if $0=\text{TEST}(P_j, M_{san})$ and $1=\text{TEST}(P_j, M_{abn_i})$ $T_{cross} \leftarrow P_j$ $M_{cross} \leftarrow \text{AD}(T_{cross})$
ROUTINE CROSSABNORMAL() $\exists i \in [1..M]$ *s.t.* $0=\text{TEST}(P_j, M_{san})$ and $0=\text{TEST}(P_j, M_{abn_i})$ $T_{cabn} \leftarrow P_j$ $M_{cabn} \leftarrow \text{AD}(T_{cabn})$

Additional Optimisation

- Data items that are indeed normal for a particular site can be considered abnormal by others.
- Proposed solution: Use a shadow server.

Performance of Collaborative Sanitisation

- Indirect model differencing performs better

Table 5. Performance when the sanitized model is poisoned and after it is cross-sanitized when using direct/indirect model differencing

Model	www1		www		lists	
	FP(%)	DR(%)	FP(%)	DR(%)	FP(%)	DR(%)
M_{pois}	0.10	44.94	0.27	51.78	0.25	47.53
M_{cross} **(direct)**	**0.24**	**100**	**0.71**	**100**	**0.48**	**100**
M_{cross} **(indirect)**	**0.10**	**100**	**0.26**	**100**	**0.10**	**100**

- Size of the cross sanitised model decreases, increasing FP rates.
- Potential attack by an adversarial collaborator.

Table 6. Size of the sanitized model when poisoned and after cross-sanitization when using direct/indirect model differencing

Model	www1		www		lists	
	#grams	file size	#grams	file size	#grams	file size
M_{abn}	2,289,888	47M	199,011	3.9M	6,025	114K
M_{pois}	1,160,235	23M	1,270,009	24M	43,768	830K
M_{cross} (direct)	**1,095,458**	**21M**	**1,225,829**	**24M**	**37,113**	**701K**
M_{cross} (indirect)	**1,160,004**	**23M**	**1,269,808**	**24M**	**43,589**	**828K**

Table 7. Time to cross-sanitize for direct and indirect model differencing

Method	www1	www	lists
direct	13.98s	26.35s	16.84s
indirect	1966.68s	1732.32s	685.81s

Polymorphic Attacks

- A polymorphic engine CLET was used to generate shellcode.
- 2100 samples of shellcode was used. 100 micro-models were poisoned with 20 shellcodes. Sanitised model was poisoned with the remaining 100 shellcode.
- 82% of the grams from 100 samples were found abnormal.