Adversarial Risk Analysis Against Malware Obfuscation Attacks (AROA)

Alberto Redondo, David Ríos

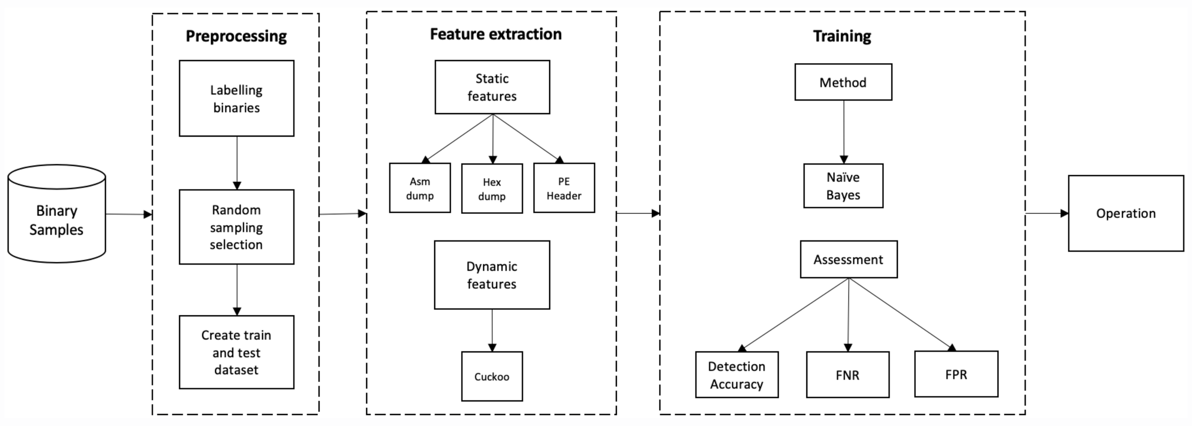
ICMAT-CSIC

Introduction

- What is **Malware Obfuscation**?
  It is malware modified in order to make it difficult to detect it
- Malware **camouflage progression**:

No Stealth	Encrypted	Oligomorphic	Polymorphic	**Metamorphic**
1970	1987	1990	1990	1998 / now

Initial Framework

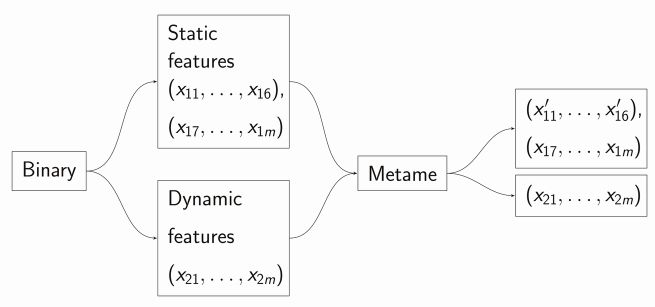


	VxHeaven			Virus Total		
Method	DA	FPR	FNR	DA	FPR	FNR
NB	0.93	0.11	0.02	0.86	0.21	0.03

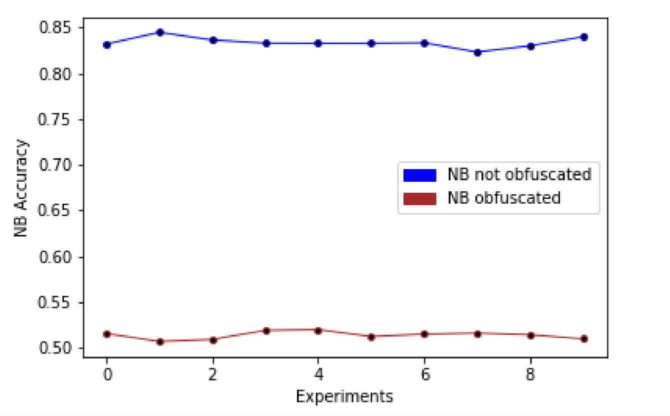
- O'Kane et al. (2016) achieved 86% DA with VxHeaven, **we obtain** 7% **DA using Naive Bayes (NB)**
- With Virus Total dataset (2018), NB decrease 8% DA with an increment of 13% FPR

Problem with obfuscated malware

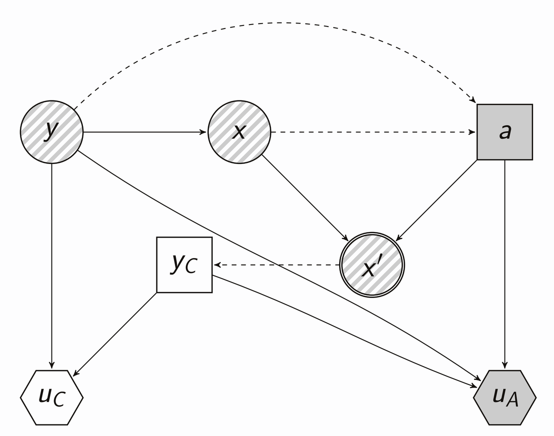
- An adversary may **obfuscate malware** affecting the accuracy of the approach
- **Metame** modifies **static features** of the binary keeping its behaviour



- **NB** is not able to detect obfuscated malware



AROA

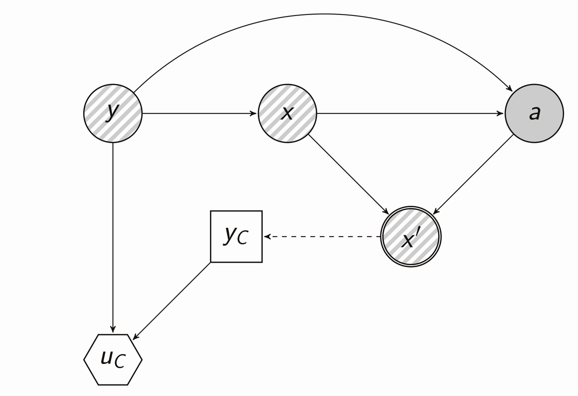


- The model adopts the **ACRA** approach, Naveiro et al. (2019)
- The problem faced by **Alan** (the adversary) and **Cleo** (the classifier) is represented through **Bi-agent influence diagram**
- **Grey nodes** are the adversary decisions
- **White nodes** are the classifier decisions
- **y**, the original class of the binary **(M=Malware, B=Benign)**
- **x**, **x′** represent the binary and the binary attacked, respectively
- **a**, Alan's attack chosen
- $\mathbf{y_C}$, Cleo's label prediction
- $\mathbf{u_C}$, $\mathbf{u_A}$ are Cleo's and Alan's utilities, respectively

Cleo's problem

Cleo's elements are:

- $p_C(y)$, with $p_C(M) + p_C(B) = 1$ and $p_C(M), p_C(B) \geq 0$
- $p_C(x|y)$
- $p_C(x'|a, x)$
- $u_C(y_C, y)$
- $\mathbf{p_C(a|x, y)}$



Cleo aims at finding the class $c(x')$ maximising her expected utility

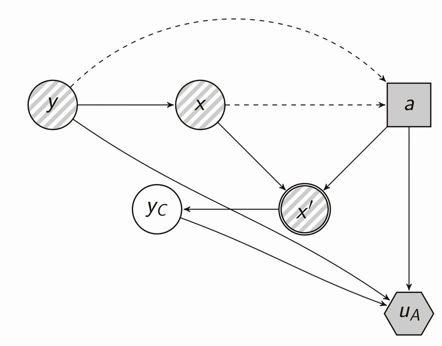
$$c(x') = \operatorname*{argmax}_{y_C} \Big[u_C(y_C, M) p_C(M) \sum_{x \in \mathcal{X}'} p_C(a_{x \to x'} | x, M) p_C(x|M) + u_C(y_C, B) p_C(x'|B) p_C(B) \Big]$$

where $\mathbf{p_C(a_{x\to x'}|x, M)}$ models the probability that Alan will perform attack $\mathbf{a_{x\to x'}}$ transforming $\mathbf{x}$ into $\mathbf{x'}$.

Alan's problem

Alan's elements are:

- $p_A(x'|a, x)$
- $u_A(y_C, y, a)$
- $p_A(c(x')|x'))$

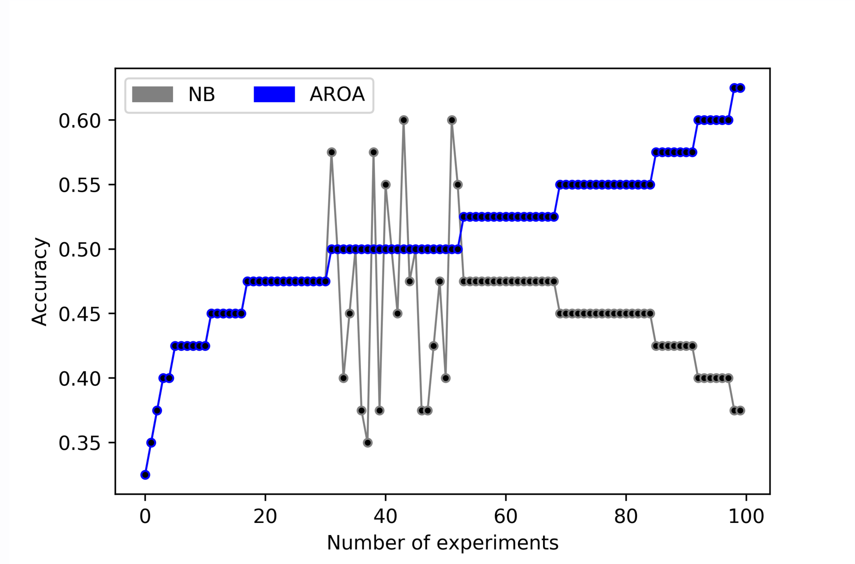


Alan seeks to maximise his expected utility through

$$A^*(x, M) = \text{argmax}_a \Big(U_A(M, M, a) - U_A(B, M, a) \Big) P^A_{a(x)} + U_A(B, M, a)$$

$\mathbf{P^A_{a(x)}}$ could be based on an estimate $\mathbf{Pr_C(c(x') = M|x')) = r}$ with $r = [0, 1]$. We could make $\mathbf{P^A_{a(x)}} \sim \beta\mathbf{e}(\delta_1, \delta_2)$.

Experimental Results



Conclusions and Ongoing work

- This approach obtains better results and describes robustness
- We are testing this approach with real data
- To advance this approach could use different obfuscation techniques, other ML algorithms or several adversaries

References

Naveiro, R., Redondo, A., Insua, D. R., and Ruggeri, F. (2019). Adversarial classification: An adversarial risk analysis approach. *International Journal of Approximate Reasoning*.

O'Kane, P., Sezer, S., and McLaughlin, K. (2016). Detecting obfuscated malware using reduced opcode set and optimised runtime trace. *Security Informatics*, 5(1):2.

Ye, Y., Li, T., Adjeroh, D., and Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. *ACM Computing Surveys (CSUR)*, 50(3):41.

{alberto.redondo,david.rios}@icmat.es