A comparison of static, dynamic, and hybrid analysis for malware detection(基于动态、静态以及混合分析方法的恶意软件检测对比))

> Damodaran A, Di Troia F, Visaggio C A, et al. A comparison of static, dynamic, and hybrid analysis for malware detection[J]. Journal of Computer Virology and Hacking Techniques, 2017, 13(1): 1-12.

> Keywords： NONE

Summary

- 比较了基于静态、动态和混合分析在恶意软件检测中的效果，最终发现纯动态的方法通常会产生最好的检测率
- 主要使用了隐马尔可夫模型（HMMs），在不同样本集进行训练和检测
- 是一篇 `survey` ，给了混合检测的一种思路，该思路主要是在训练和测试时使用了不同的特征（动态或静态），个人认为这个思路不太可行，因为样本在静态和动态时展现的行为区别还是很大的，没有可比性，果然最后结果验证了这一猜想，混合检测效果并不好
- **启发**：混合检测的方式使用特征混合的方式，而不是将动静态分别用于训练和测试，例如结合动态的API序列和静态的opcodes两种特征，同时用于训练，想来效果会不错

Glossary

- HMM：隐马尔可夫模型（Hidden Markov Model，HMM）是统计模型，它用来描述一个含有隐含未知参数的马尔可夫过程。在隐马尔可夫模型中，状态并不是直接可见的，但受状态影响的某些变量则是可见的。每一个状态在可能输出的符号上都有一概率分布。因此输出符号的序列能够透露出状态序列的一些信息。
- ROC曲线： 纵坐标：TPR 横坐标：FPR

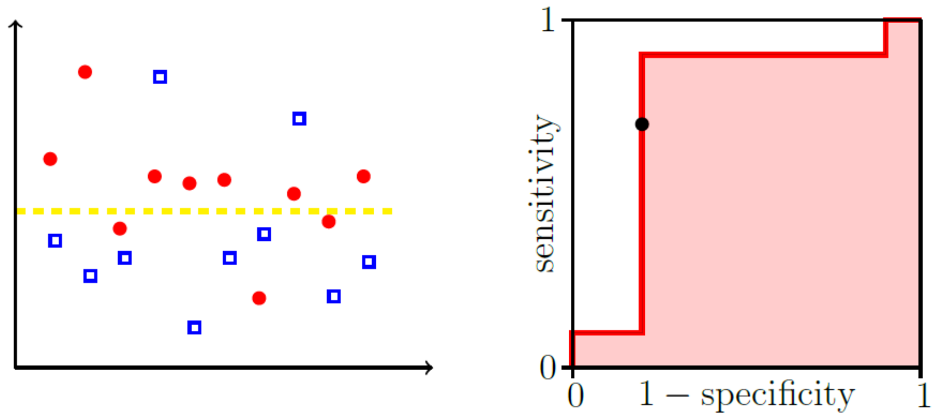


Figure 2: Scatterplot and ROC Curve

$$FPR\ (FalsePositiveRate)\ = FP/\ (FP + TN)$$

$$TPR\ (TruePositiveRate)\ = TP/\ (TP + FN)$$

- Precision & Recall

$$recall = TP/TP + FN$$

$$precision = TP/TP + FP$$

- 混淆矩阵

测试值/真实值	**P**	**N**
P	TP	FP
N	FN	TN

Research Objective(s)

- 对静态、动态和混合检测技术的缺点和优势有一定程度的了解

Background / Problem Statement

- There are many approaches to the malware detection problem.

  > Signature Based Detection、Behavior Based Detection、Statistical Based Detection

- HMM模型 可以解决一些「概率计算、解码、学习」问题

Method(s)

HMM(Hidden Markov Models)

- notation

> T = length of the observation sequence
>
> N = number of states in the model
>
> M = number of observation symbols
>
> $Q = \{q0, q1, \ldots, qN-1\}$ = distinct states of the Markov process
>
> $V = \{0, 1, \ldots, M - 1\}$ = set of possible observations
>
> A = state transition probabilities
>
> B = observation probability matrix
>
> π = initial state distribution
>
> $\mathcal{O} = (\mathcal{O}0, \mathcal{O}1, \ldots, \mathcal{O}T-1)$ = observation sequence.

- HMM模型如图

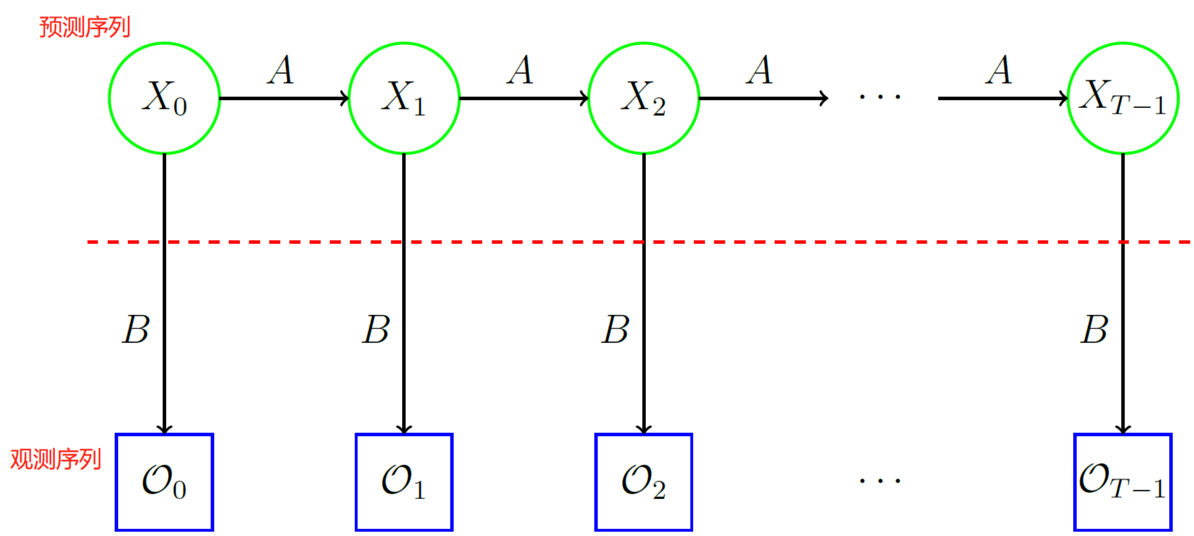


Figure 1: Generic Hidden Markov Model

Tools for Dynamic and Static Analysis

- Static Analysis:IDA Pro -- `assembly code from an executable.` 进行反汇编，获得 `opcodes` 和 `API序列`

- Dynamic Analysis:Buster Sandbox Analyzer (BSA)-- `API call sequences.`

Datasets

- 恶意样本

Table 1: Datasets

Family	Number of Files
Harebot	45
Security Shield	50
Smart HDD	50
Winwebsec	200
Zbot	200
ZeroAccess	200
benign	40

- 良性样本

Table 2: Benign Dataset

notepad	alg	calc	cipher
cleanmgr	cmd	cmdl32	driverquery
drwtsn32	dvdplay	eventcreate	eventtriggers
eventvwr	narrator	freecell	grpconv
mshearts	mspaint	netstat	nslookup
osk	packager	regedit	sndrec32
sndvol32	sol	sort	spider
syncapp	ipconfig	taskmgr	telnet
verifier	winchat	charmap	clipbrd
ctfmon	wscript	mplay32	winhlp32

- 特征示例-opcodes

`call, push, lea, push, push, call, add, test, jz`

- API序列

`OpenMutex, CreateFile, OpenProcessToken, AdjustTokenPrivileges, SetNamedSecurityInfo, LoadLibrary, CreateFile, GetComputerName, QueryProcessInformation, VirtualAllocEx, DeleteFile`

Evaluation

- AUC-ROC Results for API Call Sequence

Table 4: AUC-ROC Results for API Call Sequence

Family	Dynamic/ Dynamic	Static/ Static	Dynamic/ Static	Static/ Dynamic
Harebot	0.9867	0.7832	0.5783	0.5674
Security Shield	0.9875	1.0000	0.9563	0.8725
Smart HDD	0.9808	0.7900	0.7760	0.7325
Winwebsec	0.9762	0.9967	0.7301	0.6428
Zbot	0.9800	0.9899	0.9364	0.8879
ZeroAccess	0.9968	0.9844	0.7007	0.9106

- AUC-PR Results for API Call Sequence

Table 5: AUC-PR Results for API Call Sequence

Family	Dynamic/ Dynamic	Static/ Static	Dynamic/ Static	Static/ Dynamic
Harebot	0.9858	0.8702	0.7111	0.4888
Security Shield	0.9884	1.0000	0.9534	0.3312
Smart HDD	0.9825	0.8799	0.3768	0.4025
Winwebsec	0.9800	0.9967	0.7359	0.3947
Zbot	0.9808	0.9931	0.9513	0.3260
ZeroAccess	0.9980	0.9879	0.4190	0.3472

- ROC Results for API Call Sequence

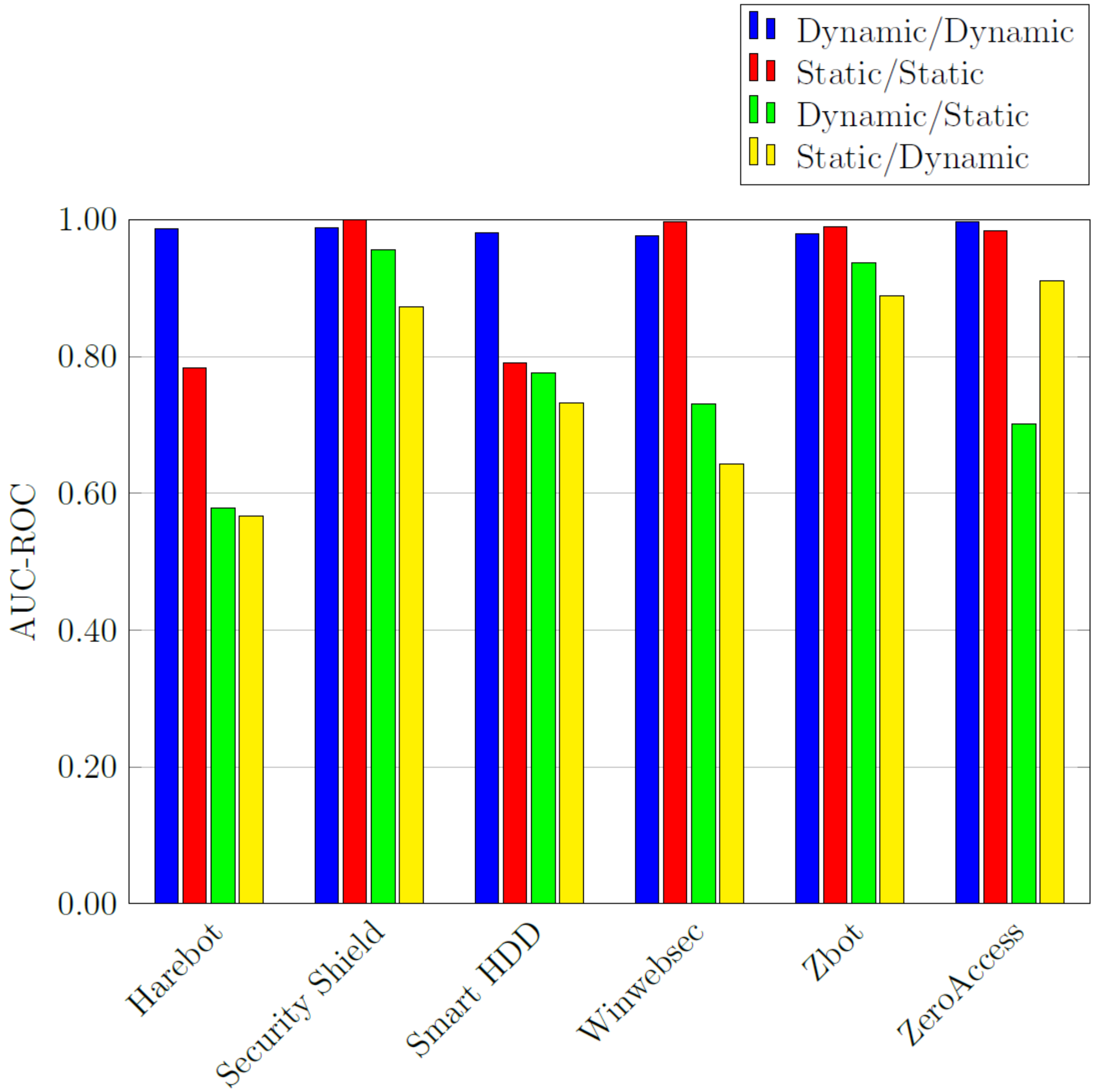


Figure 4: ROC Results for API Call Sequence

- PR Results for API Call Sequence

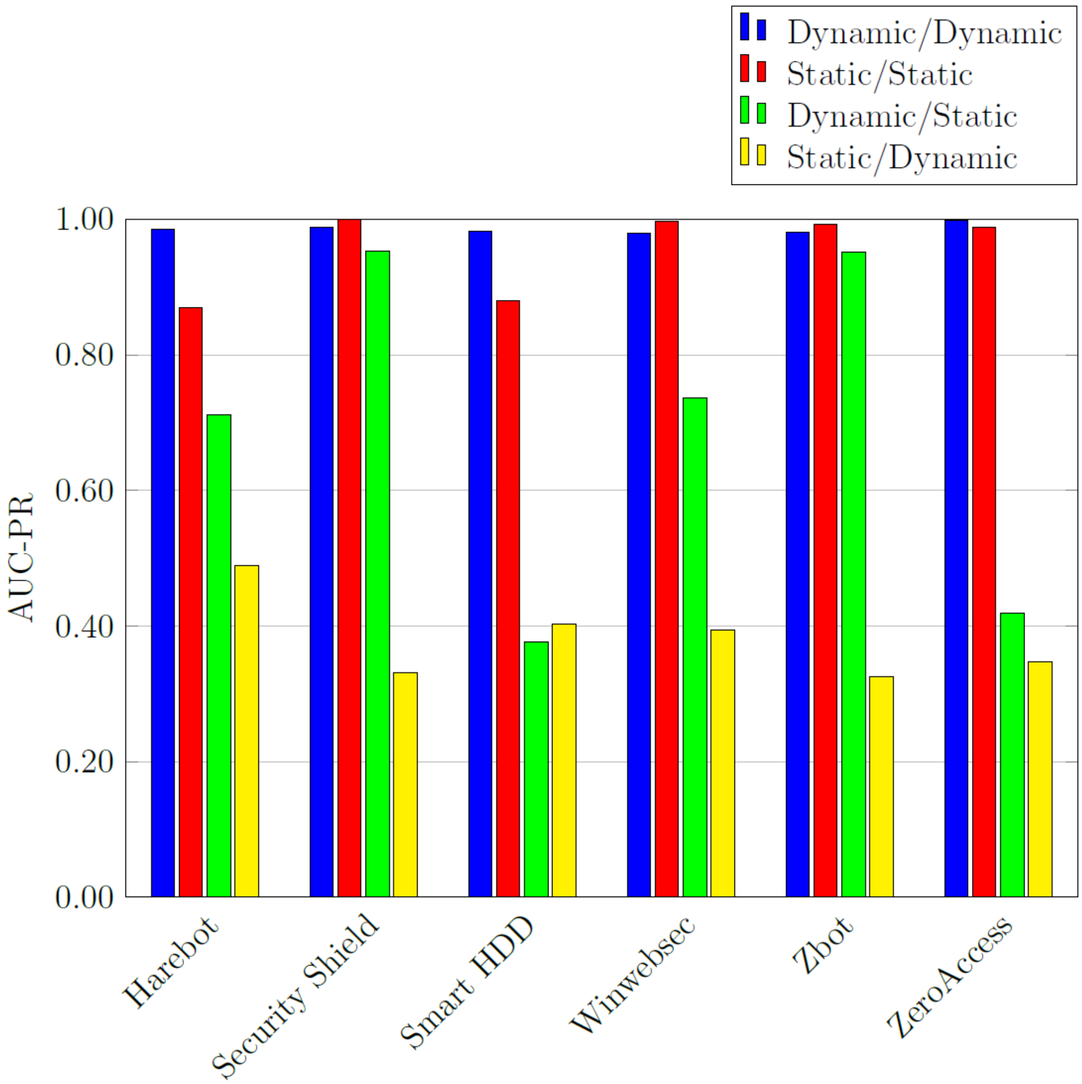


Figure 5: PR Results for API Call Sequence

- AUC-ROC Results for Opcode Sequences

Table 6: AUC-ROC Results for Opcode Sequences

Family	Dynamic/ Dynamic	Static/ Static	Dynamic/ Static	Static/ Dynamic
Harebot	0.7210	0.5300	0.5694	0.5832
Security Shield	0.9452	0.5028	0.6212	0.5928
Smart HDD	0.9860	0.9952	1.0000	0.9748
Winwebsec	0.8268	0.6609	0.7004	0.6279
Zbot	0.9681	0.7755	0.6424	0.9525
ZeroAccess	0.9840	0.7760	0.8970	0.6890

- ROC Results for Opcode Sequences

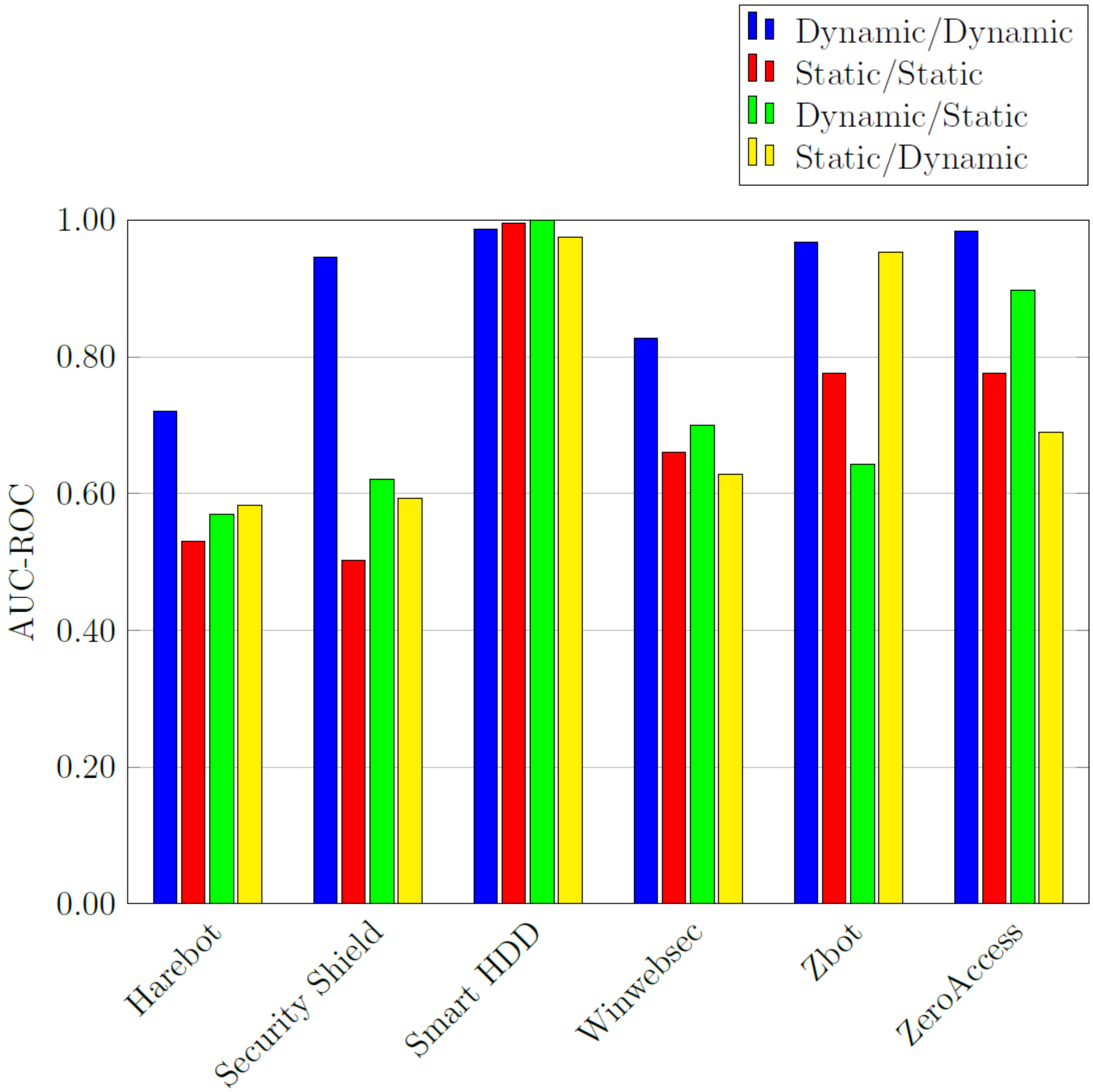
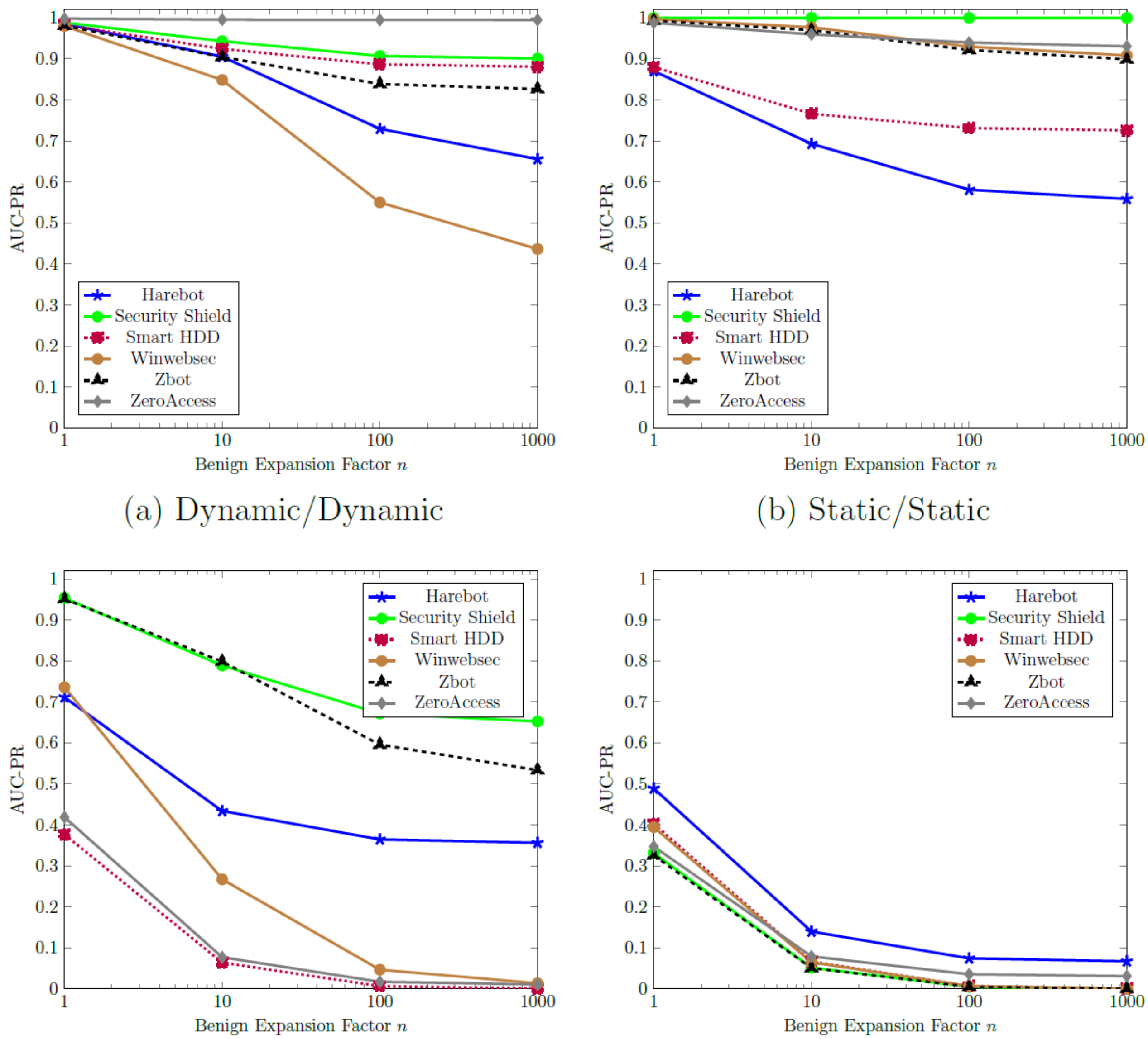


Figure 7: ROC Results for Opcode Sequences

- add factor n (模拟良性样本集数量较大的情况)



(a) Dynamic/Dynamic (b) Static/Static

(c) Dynamic/Static (d) Static/Dynamic

Figure 8: AUC-PR and Imbalanced Data (API Calls)

Conclusion

- 最终结果表明基于 API 调用的完全动态方法非常对一系列恶意软件系列有效
- 基于 API 调用的静态检测方法在大多数情况下几乎同样有效
- 混合检测方法不太有效