基於圖特徵惡意軟體檢測器的結構型對抗式樣本

Prof. Shin-Ming Cheng 鄭欣明

Connectivity Lab, CSIE, NTUST
Research Center for IT Innovation, Academia Sinica

Chi-Hsin Yang 楊季昕

IsLab, CSIE, NTUST

Shin-Ming Cheng 鄭欣明
台科大資工系 教授

- 中研院資創中心 合聘研究員
- 日本NICT Cybersecurity Lab 訪問學者, 2019
- 教育部AIS3 主持人, 2015 迄今
  - AIS3, MyFirstCTF, EOF, AIS3 Club
  - 協助帶隊參加 DEFCON, 2018 迄今
- CCoE人才培育, 2021迄今
- 5G Security –Rogue BS Attack
  - HITCON 2020, CISC 2020 Best Paper Award, CyberSec 2021, HITCON 2021 5G Village
- IoT Security –Firmware Emulation/Rehosting
  - Cybersec 2021, HITCON 2021 Workshop
- Security of AI for Cybersecurity
  - IEEE Trustcom 2020 and CISC 2021 Best Paper Award, HITCON 2021

Chi-Hsin Yang 楊季昕

- 國立臺灣科技大學 資訊工程系 碩士
- 第五、六屆臺灣好厲駭成員

Outline

✓ Introduction
- Malware Detection
- Adversarial Attack
- Challenges & Goals
- Related Work

✓ Approach

✓ Experiments Setup

✓ Evaluation

✓ Discussion

Malware Detection (1/3)

✓ Most traditional detectors can only identify known samples, and for new types of malware, it is easy to be judged as unknown samples.

✓ Attackers often use packaging, obfuscation, etc., to make them different from known threats to evade detection.

→ Detectors using **machine learning** algorithms and huge amounts of data will have the opportunity to identify unknown malware.

Malware Detection (2/3)

✓ **Static Analysis**

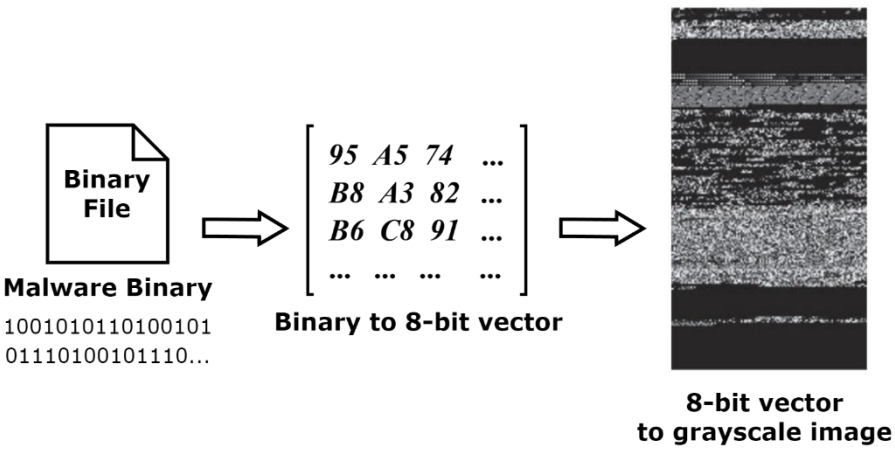
- Analysis from binary files→ **lower time cost**
- Don't need to actually execute the program

  → Has the advantage of **cross-architecture analysis**

✓ **Static Feature in Malware Analysis**

1) Binary-based

- Byte sequence, Grayscale images



E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware detection by eating a whole EXE," in Proc. AAAI 2018, Jun. 2018.

H. S. Anderson and P. Roth, "EMBER: An open dataset for training static PE malware machine learning models," arXiv preprint arXiv:1804.04637, Apr. 2018.

X. Liu, Y. Lin, H. Li, and J. Zhang, "A novel method for malware detection on ML-based visualization technique," Computers & Security, vol. 89, p. 101682, Feb. 2020.

Malware Detection (3/3)

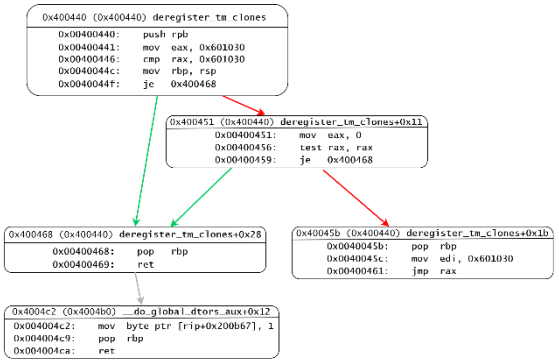
✓ **Static Feature in Malware Analysis**

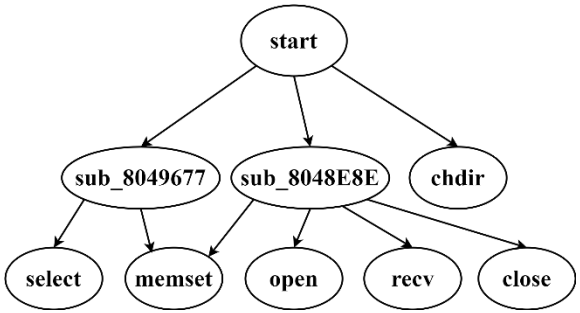
2) Signature-based

- Header Information, Printable Strings, Opcodes

3) Structure-based

- Control Flow Graph(CFG), Function Call Graph(FCG)





M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, and Z. Chen, “Efficient signature generation for classifying cross-architecture IoT malware,” in Proc. IEEE CNS 2018, May 2018.

C.-W. Tien, S.-W. Chen, T. Ban, and S.-Y. Kuo, “Machine learning framework to analyze IoT malware using ELF and opcode features,” Digital Threats: Research and Practice, vol. 1, no. 1, pp. 1–19, Mar. 2020.

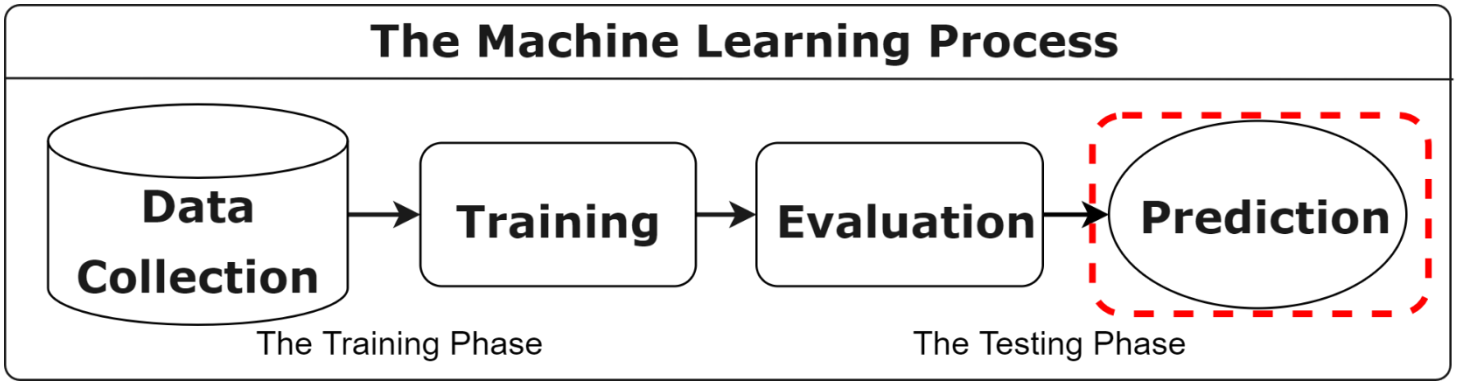
H. Alasmary, A. Khormali, A. Anwar, J. Park, J. Choi, A. Abusnaina, A. Awad, D. Nyang, and A. Mohaisen, “Analyzing and detecting emerging Internet of Things malware: A graph-based approach,” IEEE Internet Things J., vol. 6, no. 5, pp. 8977–8988, Oct. 2019.

C.-Y. Wu, T. Ban, S.-M. Cheng, B. Sun, and T. Takahashi, “IoT malware detection using function-call-graph embedding,” in Proc. IEEE PST 2021, Dec. 2021, pp. 1–9.

Adversarial Attack (1/6)

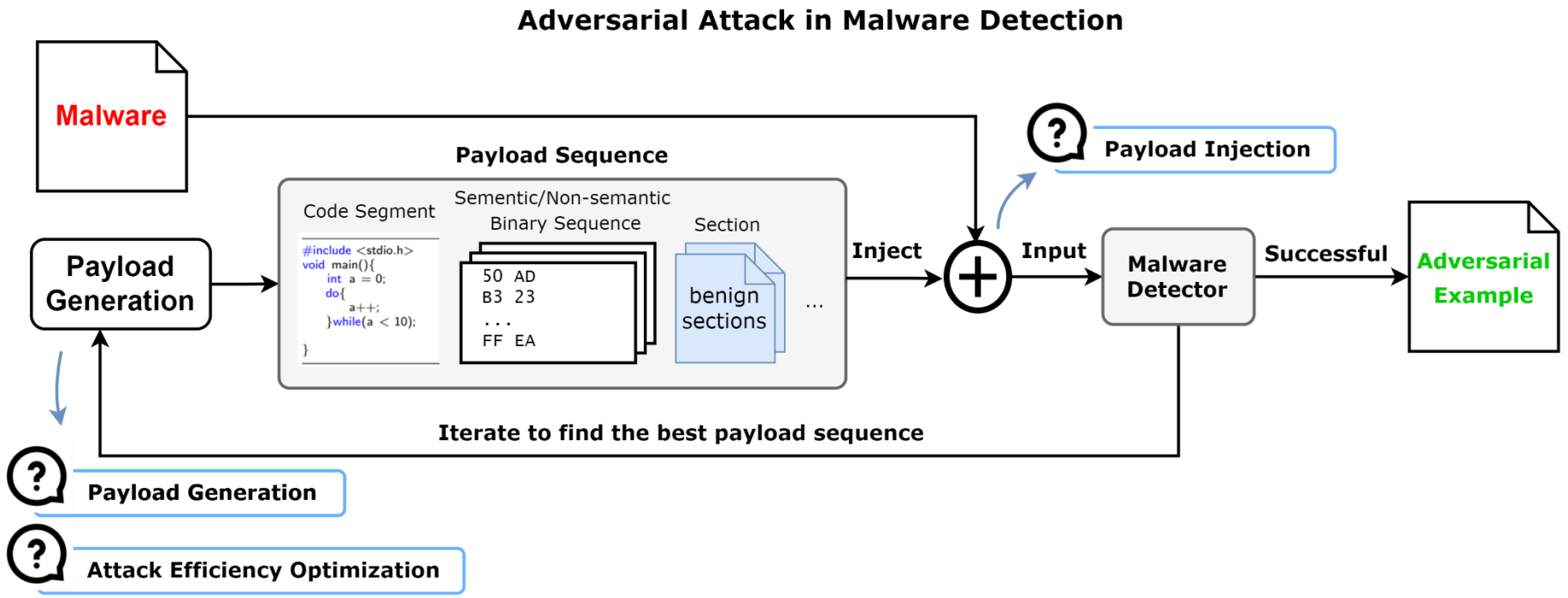
✓ Inject imperceptible perturbations into the samples to mislead the model.

✓ An effective way to assess the robustness of machine learning models.



I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, Mar. 2015.

Adversarial Attack (2/6)



L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Functionality-preserving black-box optimization of adversarial windows malware," *IEEE Trans. Inf. Forensics Security*, vol. 16,pp. 3469–3478, May 2021.

M. Ebrahimi, N. Zhang, J. Hu, M. T. Raza, and H. Chen, "Binary black-box evasion attacks against deep learning-based static malware detectors with adversarial byte-level language model," Feb. 2021.

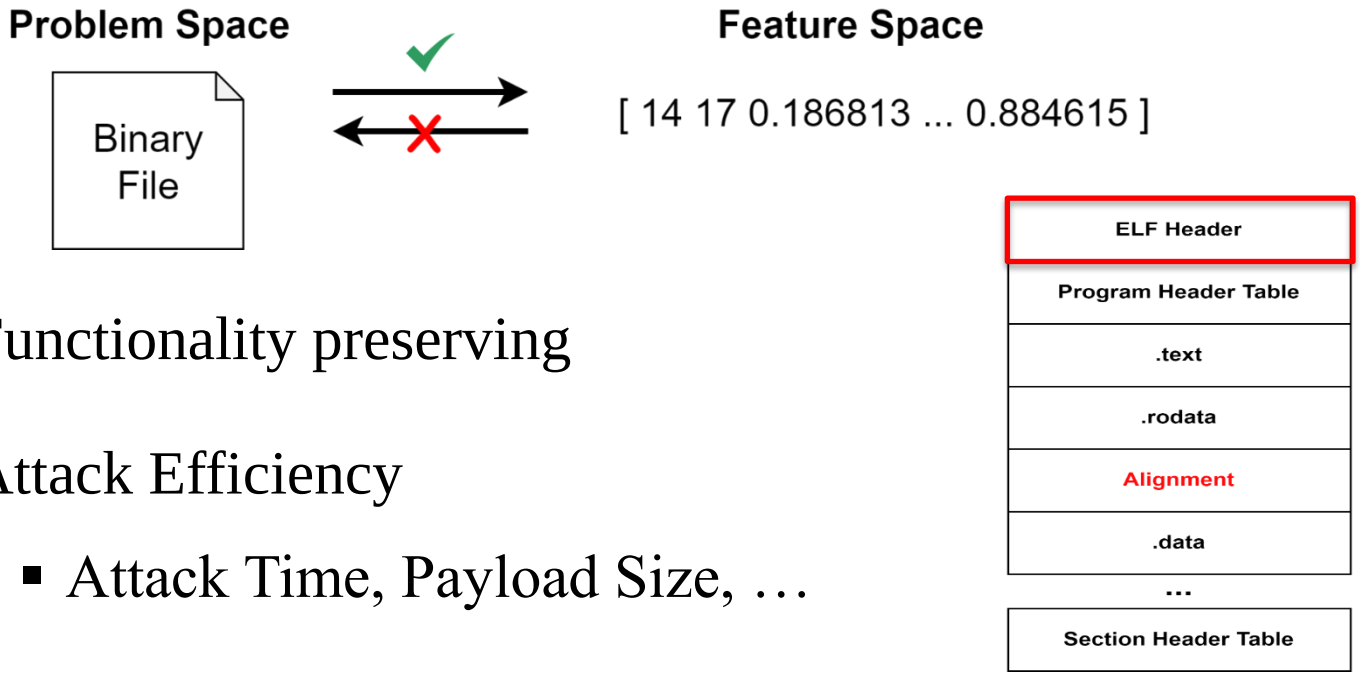
C. Yang, J. Xu, S. Liang, Y. Wu, Y. Wen, B. Zhang, andD. Meng, "DeepMal: maliciousness-preserving adversarial instruction learning against static malware detection,"Cybersecurity, vol. 4, May 2021.

L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando,and F. Roli, "Adversarial EXEmples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection, "*ACM Trans. Privacy and Security*, vol. 24,no. 4, pp. 1–31, Nov. 2021

Adversarial Attack (3/6)

✓ Challenge of Adversarial Attack in Malware Detection

- Available Transformation



- Functionality preserving
- Attack Efficiency
  - Attack Time, Payload Size, …

F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro.,"Intriguing properties of adversarial ML attacks in the problem space," in *Proc. IEEE S&P 2020*, May 2020, pp. 1332–1349

K. Lucas, M. Sharif, L. Bauer, M. K. Reiter, and S. Shintre,"Malware makeover: Breaking ML-based static analysis by modifying executable bytes," in *Proc. ACM Asia CCS 2021*, May2021, pp. 744–758.

Adversarial Attack (4/6)

✓ **Functionality preserving Adversarial Attack**

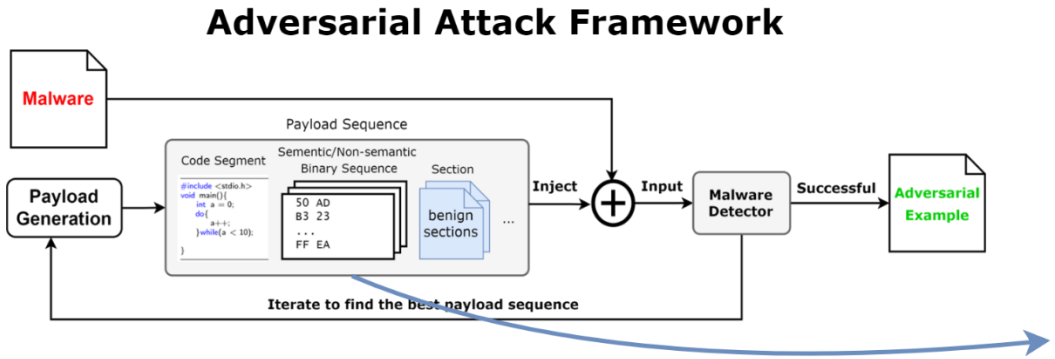
- Code-level Payload Injection and Generation
- Binary-level Payload Injection
  - Meaningless Payload Generation
  - **Meaningful Payload Generation**

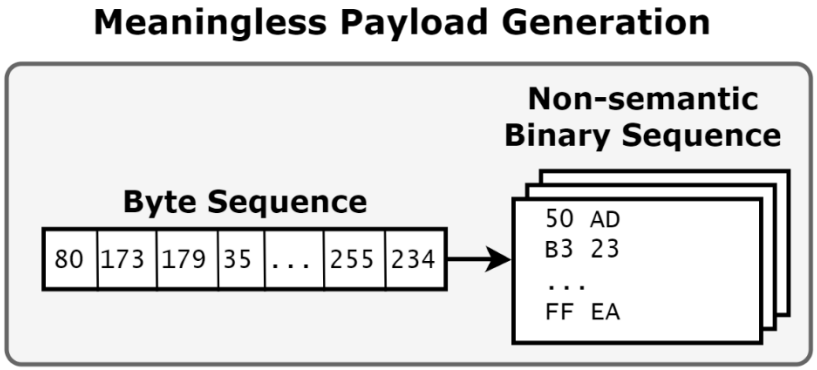
✓ **Code-level Payload Injection and Generation**

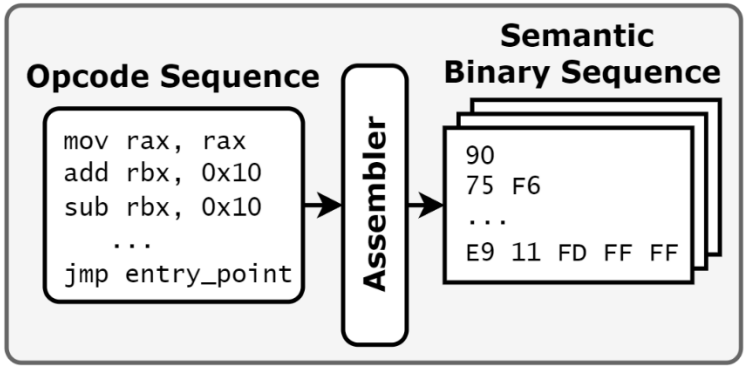
- **Target Detector:** Opcode- and structure-based detector
- **Method:** Code Injection (Abusnaina *et al.*), Rewrite Code (Zhao *et al.*)

A. Abusnaina, A. Khormali, H. Alasmary, J. Park, A. Anwar, and A. Mohaisen, "Adversarial learning attacks on graph-based IoT malware detection systems," in Proc. IEEE ICDCS 2019, Jul. 2019, pp. 1296– 1305.
K. Zhao, H. Zhou, Y. Zhu, X. Zhan, K. Zhou, J. Li, L. Yu, W. Yuan, and X. Luo, "Structural attack against graph based android malware detection," in Proc. ACM Asia CCS 2021, Nov. 2021, p. 3218–3235.

Adversarial Attack (5/6)







Adversarial Attack (6/6)

✓ **Binary-level Payload Injection**

- **Meaningless Payload Generation**
  - **Target Detector:** Only opcode-based detector
  - **Method:** Padding (Demetrio *et al.*), Content Shifting(Demetrio *et al.*)
- **Meaningful Payload Generation**
  - **Target Detector:** Opcode- and structure-based detector
  - **Method:** Inject/Rewrite Opcode (Lucas *et al.*), Semantic NOPs (Zhang *et al.*)

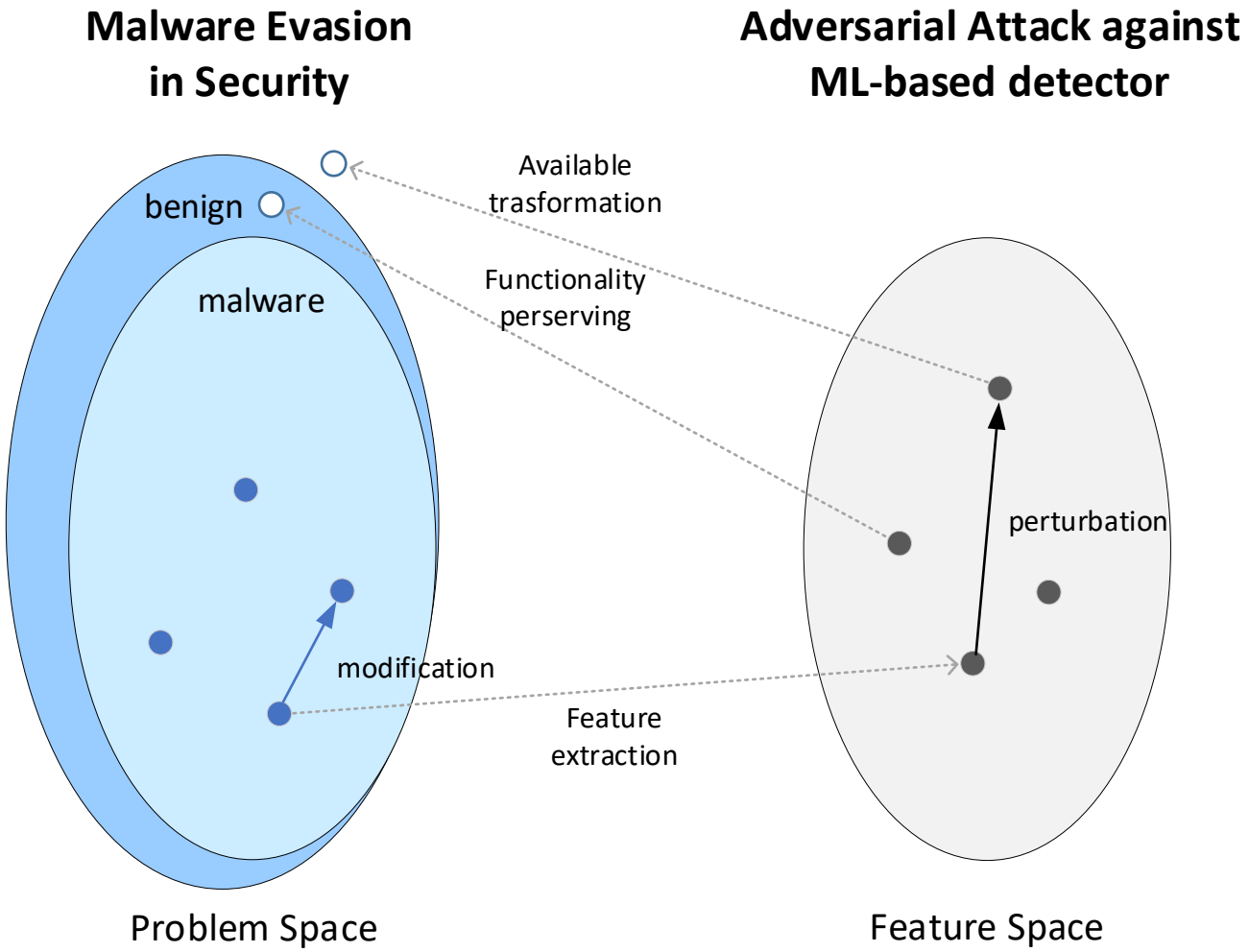
L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Functionality-preserving black-box optimization of adversarial windows malware," IEEE Trans. Inf. Forensics Security, vol. 16, pp. 3469– 3478, May 2021.

L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando, and F. Roli, "Adversarial EXEmples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection," ACM Trans. Privacy and Security, vol. 24, no. 4, pp. 1–31, Nov. 2021.

K. Lucas, M. Sharif, L. Bauer, M. K. Reiter, and S. Shintre, "Malware makeover: Breaking ML-based static analysis by modifying executable bytes," in Proc. ACM Asia CCS 2021, May 2021, pp. 744–758.

L. Zhang, P. Liu, Y.-H. Choi, and P. Chen, "Semantics-preserving reinforcement learning attack against graph neural networks for malware detection," IEEE Trans. Dependable Secure Comput., Mar. 2022.

Challenges & Goals (1/2)



Challenges & Goals (2/2)

✓ **Challenges**

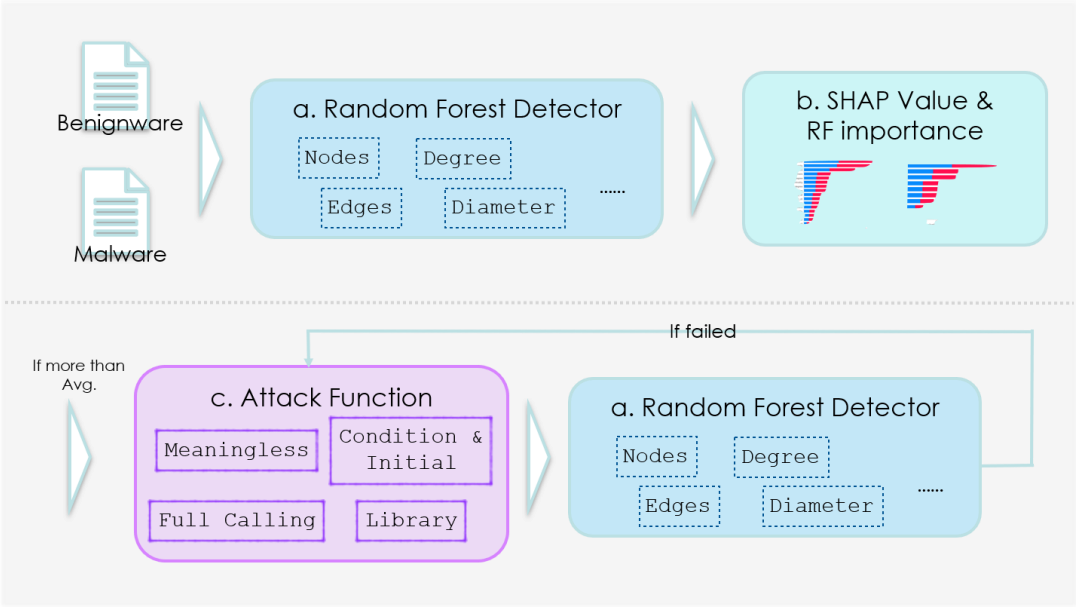
- Available transformation
- Functionality preserving
- Attack Efficiency

✓ **Goals**

- Successfully attack the structure-based malware detector
- Effectively reduce the attack cost
- Compare and analyze the robustness of different detectors

Related Work (1/5)

Paper	Robustness Evaluation of Graph-based Malware Detection Using Code-leel Adversarial Attack with Explainability
Method	▪ Code-level attack ▪ Evaluating the robustness of CFG-based detectors that using structure features and hybrid structure and opcode features.
Summary	▪ Their results confirm that **detectors with both CFG structure and Opcode features are more robust** than detectors using only CFG structure features.

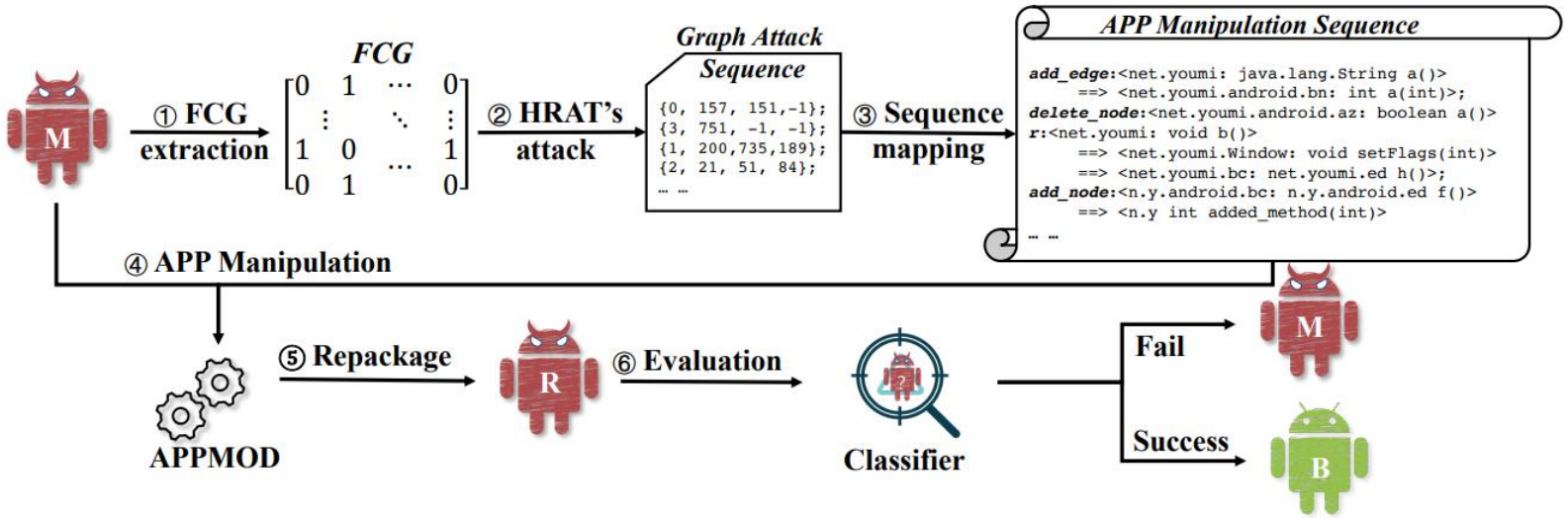


Structural	Opcode
Nodes	Total_trans
Edges	Total_cal
Degree	Total_ctl
Degree	Avg_trans
Density	Avg_cal
Closeness_cent	Avg_ctl
Betweenness_cent	Avg_block_size
Connected_com	
Diameter	
Radius	
Avg_block	

L.-B. Ouyang, “Robustness evaluation of graph-based malware detection using code-level adversarial attack with explainability,” Master, NTUST, Taipei, Taiwan, Jul. 2021.

Related Work (2/5)

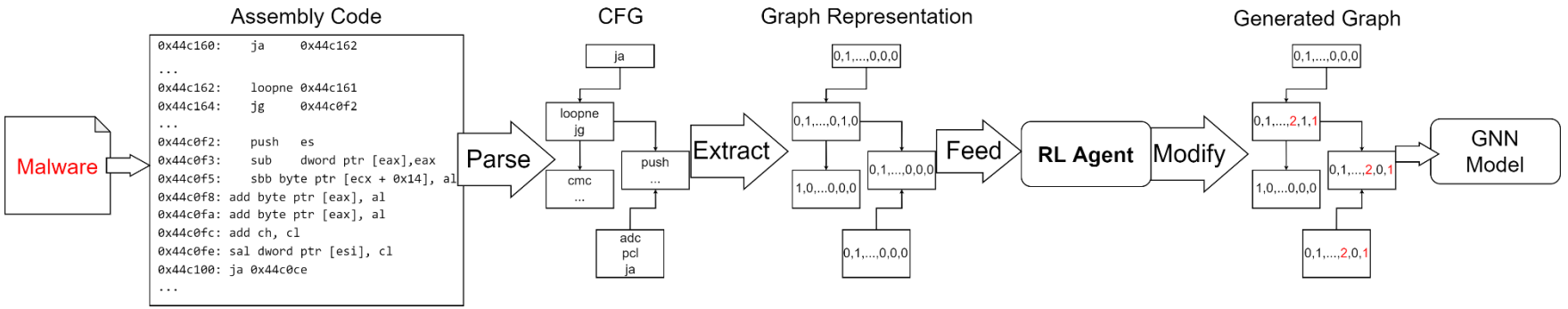
Paper	Heuristic optimization model integrated with Reinforcement learning framework to optimize our structural ATtack (HRAT)
Method	▪ Code-level attack ▪ Use reinforcement learning(RL) to choose where to modify the FCG and rewrite the code according to the available transformations they proposed.
Summary	▪ Their attack connects feature space to problem space, and achieves a high success rate on Android malware detectors.
Limitation	▪ High computational consumption



K. Zhao, H. Zhou, Y. Zhu, X. Zhan, K. Zhou, J. Li, L. Yu, W. Yuan, and X. Luo, "Structural attack against graph based android malware detection," in Proc. ACM Asia CCS 2021, Nov. 2021, p. 3218–3235.

Related Work (3/5)

Paper	Semantics-preserving Reinforcement Learning Attack Against Graph Neural Networks for Malware Detection[52]
Method	▪ Binary-level/ meaningful attack ▪ Using RL to iteratively chooses basic blocks and semantic nops, then modifying the malicious CFG to generate the adversarial example.
Summary	▪ Their meaningful attack achieves a higher evasion rate on GNN malware detectors.
Limitation	▪ Unable to affect structure-based models



Semantic Nops

NOP	
PUSH %rbx	POP %rbx
NOT %rbx	NOT %rbx
XCHG %rax,%rax	XCHG %rax,%rax
ADD $5,%r10	SUB $5,%r10

L. Zhang, P. Liu, Y.-H. Choi, and P. Chen, "Semantics-preserving reinforcement learning attack against graph neural networks for malware detection," IEEE Trans. Dependable Secure Comput., Mar. 2022.

Related Work (4/5)

Paper	Malware Makeover: Breaking ML-based Static Analysis by Modifying Executable Bytes
Method	▪ Binary-level/ meaningful attack ▪ Using in-place randomization and code displacement to attack binary-based detectors.
Summary	▪ Their meaningful attack achieves a higher evasion rate on binary-based detectors.
Limitation	▪ Has little effect on structure-based models

In-place Randomization

(a) Original		(b) Equivalent instructions		(c) Register reassignment		(d) Instruction reordering		(e) Register preservation	
push ebp	(55)	push ebp	(55)	push ebp	(55)	push ebp	(55)	push ebp	(55)
mov ebp, esp	(89e5)	mov ebp, esp	(89e5)	mov ebp, esp	(89e5)	mov ebp, esp	(89e5)	mov ebp, esp	(89e5)
push ebx	(53)	push ebx	(53)	push ebx	(53)	push ebx	(53)	**push edx**	**(52)**
push edx	(52)	push edx	(52)	push edx	(52)	push edx	(52)	**push ebx**	**(53)**
mov ebx, [ebp+4]	(8b5d04)	mov ebx, [ebp+4]	(8b5d04)	**mov edx, [ebp+4]**	**(8b5504)**	**mov ebx, [ebp+8]**	**(8b5d08)**	mov ebx, [ebp+8]	(8b5d08)
add ebx, 0x10	(83c310)	**sub ebx, -0x10**	**(83ebf0)**	**sub edx, -0x10**	**(83eaf0)**	**mov edx, [ebp+4]**	**(8b5504)**	mov edx, [ebp+4]	(8b5504)
mov edx, [ebp+8]	(8b5508)	mov edx, [ebp+8]	(8b5508)	**mov ebx, [ebp+8]**	**(8b5d08)**	**sub edx, -0x10**	**(83eaf0)**	sub edx, -0x10	(83eaf0)
mov [edx], ebx	(891a)	mov [edx], ebx	(891a)	**mov [ebx], edx**	**(8913)**	mov [ebx], edx	(8913)	mov [ebx], edx	(8913)
pop edx	(5a)	pop edx	(5a)	pop edx	(5a)	pop edx	(5a)	**pop ebx**	**(5b)**
pop ebx	(5b)	pop ebx	(5b)	pop ebx	(5b)	pop ebx	(5b)	**pop edx**	**(5a)**
pop ebp	(5d)	pop ebp	(5d)	pop ebp	(5d)	pop ebp	(5d)	pop ebp	(5d)

K. Lucas, M. Sharif, L. Bauer, M. K. Reiter, and S. Shintre, "Malware makeover: Breaking ML-based static analysis by modifying executable bytes," in Proc. ACM Asia CCS 2021, May 2021, pp. 744–758.

Related Work (5/5)

Method	Target Detector	Payload Generation	Payload Injection	Attack Efficiency Optimization
Ouyang *et al.*	Structure, Signature	Code	Code	Feedback of Explainability Method
Zhao *et al.*	Structure	Code	Code	Reinforcement Learning (RL)
Zhang *et al.*	Structure, Signature	Meaningful	Binary	Reinforcement Learning (RL)
Lucas *et al.*	Binary	Meaningful	Binary	Hill-Climbing Algorithm (HC)
Our proposed	Structure, Signature	Meaningful	Binary	Hill-Climbing Algorithm (HC), Optimization Algorithm (OA)

Outline

✓ Introduction

✓ Approach

- Thread Model
- Framework
- Payload Generation
- Attack Strategy

✓ Experiments Setup

✓ Evaluation

✓ Discussion

Threat Model (1/1)

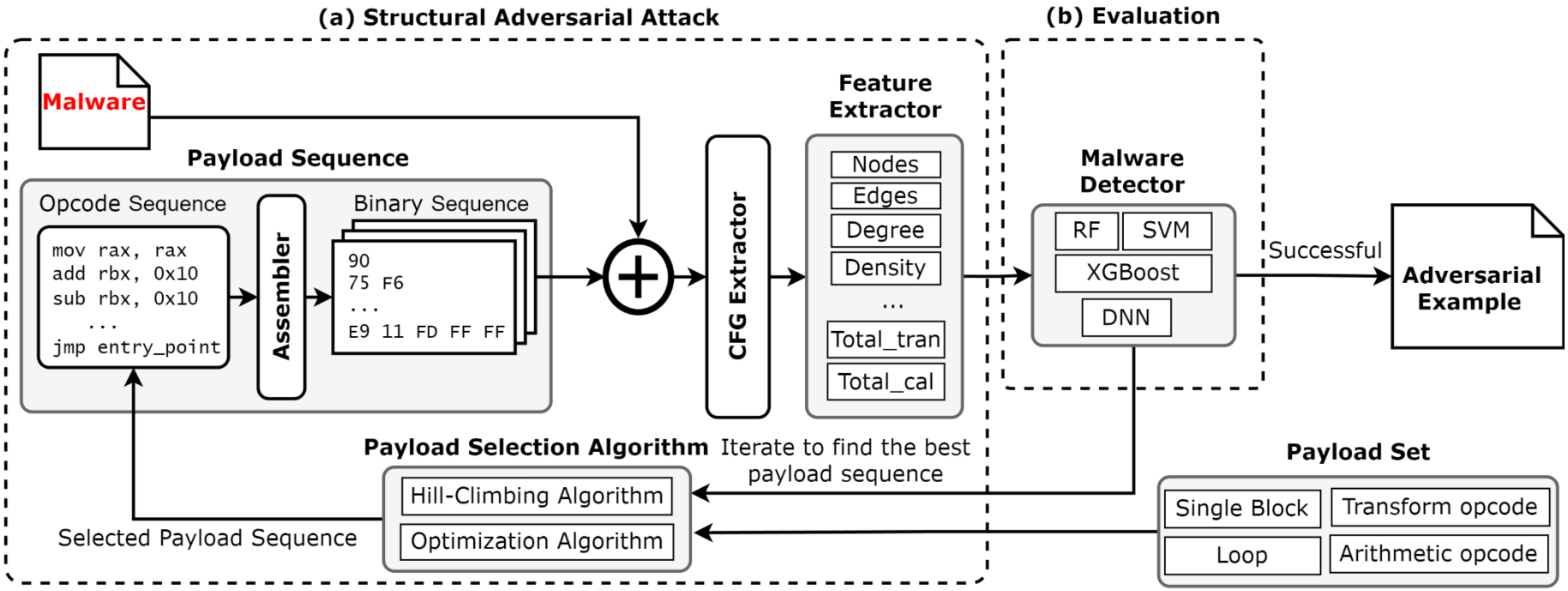
✓ **Adversary's Capability**

- Adversary has **black-box** access to the target model.

✓ **Adversary's Goal**

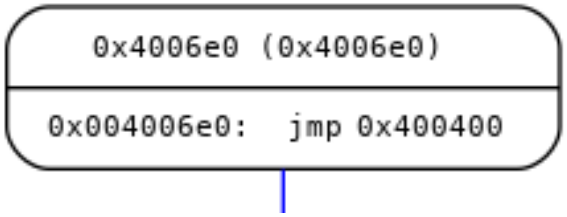
- Generate malware samples that can mislead malware detectors into classifying them as benign.
- Functionality preserving modification
  - Ensured that the sample **can still execute and retain the original malicious function.**

Framework (1/1)

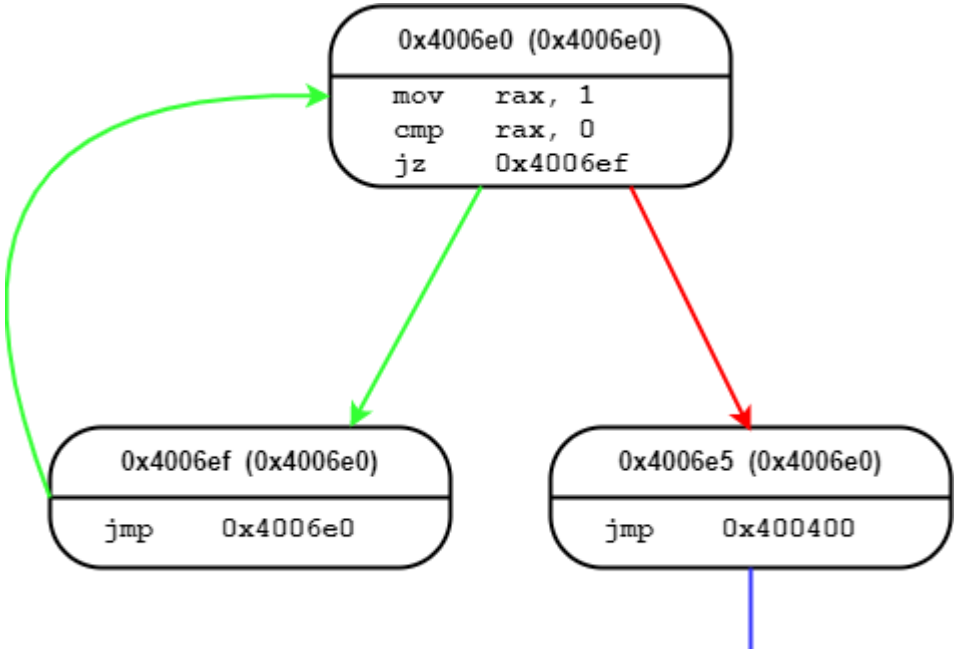


Payload generation (1/2)

✓ **Single block**



✓ **Loop**



T.-Y. Chen, "Structural attack against graph-based IoT malware detection at assembly level," Master, NTUST, Taipei, Taiwan, Jan. 2022.

Payload generation (2/2)

✓ **Transform Opcode**

- ✓ push
- ✓ pop
- ✓ mov
- ✓ movabs
- ✓ lui

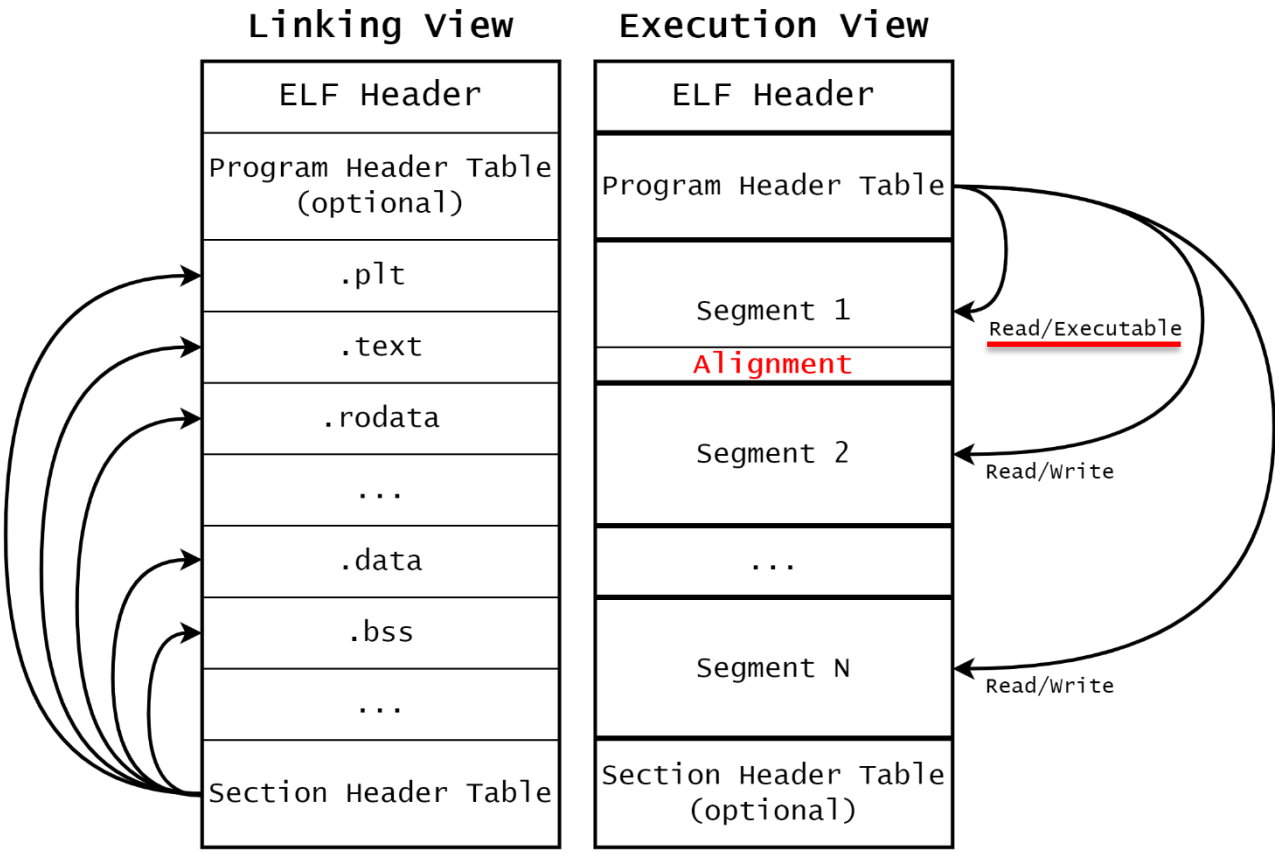
✓ **Arithmetic Opcode**

- ✓ add
- ✓ sub
- ✓ xor
- ✓ and
- ✓ div

T.-Y. Chen, “Structural attack against graph-based IoT malware detection at assembly level,” Master, NTUST, Taipei, Taiwan, Jan. 2022.

Attack Strategy (1/2)

✓ **Code Cave**



Attack Strategy (2/2)

✓ **Hill-Climbing Algorithm**

- Inject four payloads, respectively
- Choose the payload sequence that reduces the most confidence

✓ **Optimization Algorithm**

- Random parameters give the system a chance of not accepting the current solution → **Helping to escape the local minimum**
- Dynamically adjust the amount of payload injected according to changes in confidence → **Helping reduce iterations**

Outline

✓ Introduction

✓ Approach

✓ Experiments Setup

- Datasets & Target Detectors

✓ Evaluation

✓ Discussion

Datasets & Target Detectors (1/1)

Features and categories of features used by malware detectors

Structural	Opcode
Nodes	Total_trans
Edges	Total_cal
Out_degree	Total_ctl
In_degree	Avg_trans
Density	Avg_cal
Closeness_cent	Avg_ctl
Betweenness_cent	Avg_block_size
Connected_com	
Diameter	
Radius	
Avg_block	

Distribution of IoT sample in the dataset

Class	Train	Test	Total
Benign	6507	1649	8156
Malicious	4949	1215	6164

Accuracy of detectors with different settings

Detector	Feature	Accuracy Train(%)	Accuracy Test(%)
RF	**Structure**	99.55	97.06
	Opcode	99.98	98.84
	Hybrid	99.98	98.97
SVM	**Structure**	98.16	96.06
	Opcode	99.91	97.83
	Hybrid	99.92	98.47
XGBoost	**Structure**	99.02	96.06
	Opcode	99.50	98.72
	Hybrid	99.78	99.11
DNN	**Structure**	95.80	95.43
	Opcode	96.48	95.17
	Hybrid	98.12	98.35

L.-B. Ouyang, “Robustness evaluation of graph-based malware detection using code-level adversarial attack with explainability,” Master, NTUST, Taipei, Taiwan, Jul. 2021.

Outline

✓ Introduction

✓ Approach

✓ Experiments Setup

✓ Evaluation

- Attack Results
- Adversarial Example Transferability

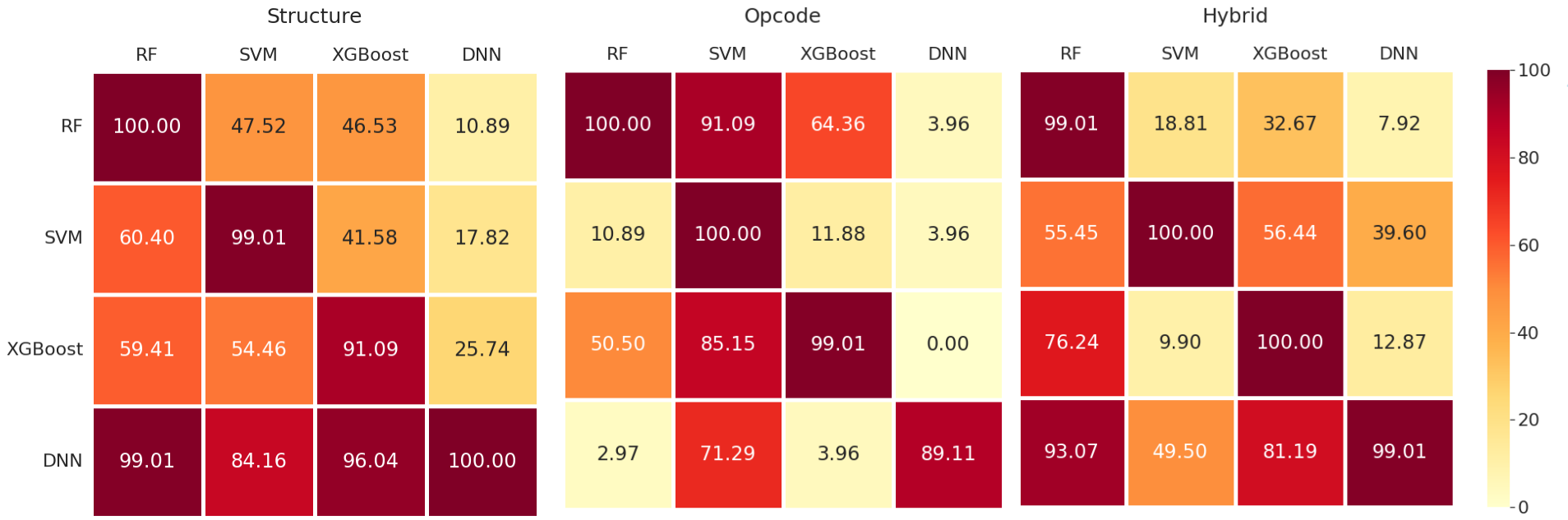
✓ Discussion

Attack Results (1/1)

HC: Hill-Climbing Algorithm
OA: Optimization Algorithm

Detector feature	Attack Algorithm	Detector	Evasion Rate(%)	Append Size(%)	Iterations
Structure	HC	RF	98.02	**11.58**	48.07
		SVM	96.04	**9.73**	44.30
		XGBoost	**91.09**	11.55	54.57
		DNN	100.00	18.14	83.47
	OA	RF	**100.00**	12.31	**15.03**
		SVM	**99.01**	10.75	**15.13**
		XGBoost	**91.09**	**9.48**	**12.45**
		DNN	**100.00**	**14.84**	**23.29**
Opcode	HC	RF	**100.00**	**33.88**	157.16
		SVM	**100.00**	10.45	49.53
		XGBoost	**99.01**	27.68	132.55
		DNN	88.12	38.18	184.78
	OA	RF	**100.00**	34.68	**44.46**
		SVM	**100.00**	**9.46**	**22.40**
		XGBoost	**99.01**	**27.53**	**34.09**
		DNN	**89.11**	**36.03**	**30.23**
Hybrid	HC	RF	**99.01**	18.09	83.74
		SVM	**100.00**	12.08	63.65
		XGBoost	**100.00**	20.38	93.85
		DNN	**99.01**	24.31	118.53
	OA	RF	**99.01**	**13.79**	**19.54**
		SVM	**100.00**	**10.85**	**23.54**
		XGBoost	**100.00**	**12.67**	**18.24**
		DNN	**99.01**	**17.66**	**30.71**

Adversarial Example Transferability (1/1)



✓ The sample transfer rate is higher between models of the same type.

✓ Tree-based models that use opcode-based features, and SVM or DNN models that use hybrid features are more robust.

✓ Overall, **DNN is the most robust model,** and the model using only structure-based features has less robustness.

Outline

✓ Introduction

✓ Approach

✓ Experiments Setup

✓ Evaluation

✓ Discussion

- Conclusions
- Future Work

Conclusions (1/1)

✓ Functionality preserving

✓ Meaningful attack

✓ Our attack receives a high evasion rate with a **lower attack cost.**

✓ The adversarial example generated by our attack has **transferability**.

Future Work (1/1)

✓ We would like to extend this attack system to detectors and even classifiers trained on different features.

✓ We would like to exploit **explainability methods** to attack black-box models accurately.

Thank You
Q&A