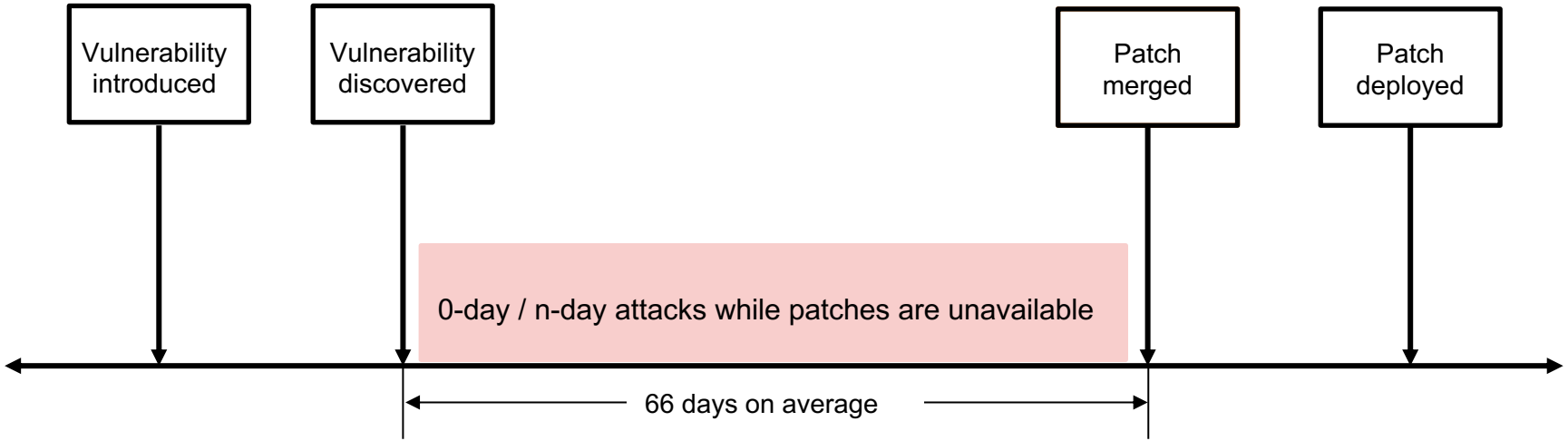
On-the-fly Quarantine Before Patches for N-day Kernel Vulnerabilities Are Available

Zicheng Wang, Tiejin Chen, Qinrun Dai, Yinggang Guo, Yueqi Chen, Hua Wei

Work-in-Progress

Background

- N-day Vulnerabilities

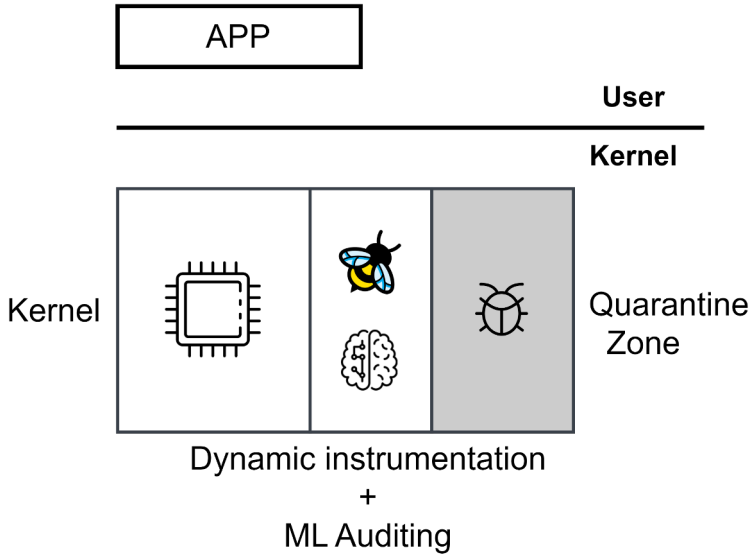


Real-time Defense

- Usenix Security’23: PET, Prevent Errors From Being Triggered.
- Usenix Security’24: SeaK, Prevent Heap Vulnerabilities From Being Exploited
- Vulnerability Behaviors are Complex
  - multiple triggering condition / exploitation path
  - current defense are ad-hoc

On-the-fly OS Quarantine (O2Q)

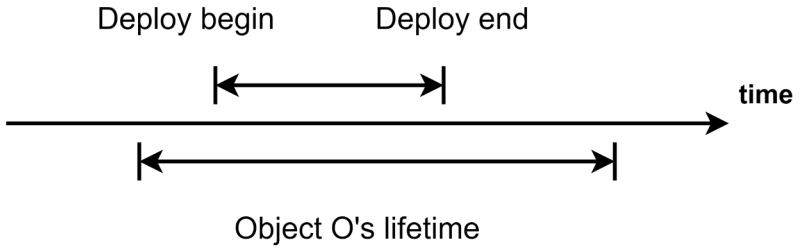
- **Eliminate the complex vulnerabilities inside the Quaratine Zone**
- Classic sandbox can absolutely isolate complex vulnerabilites
- Design for 0-day vulnerabilities, hardly deploy on-the-fly compare to the related work
- Challenge: Object-lifetime problem

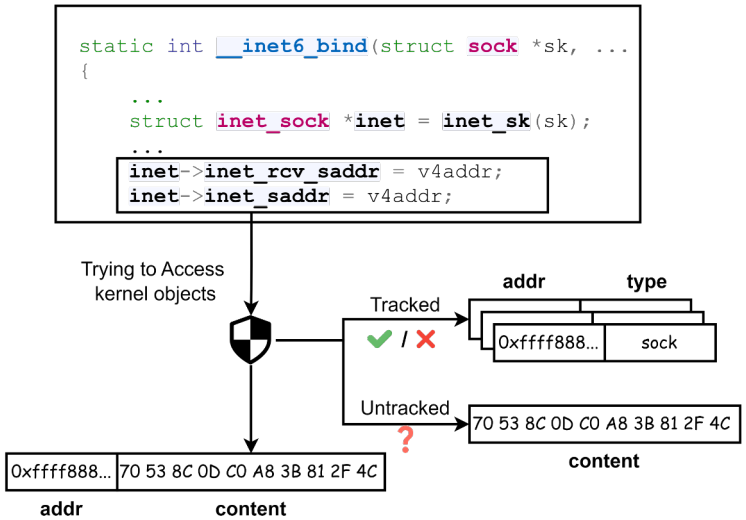


Object-lifetime Problem

- Object O belongs to Quarantine Q
- O was allocated before the deployment
- O is not released after deployment
- O is not tracked by the sandbox
- O has no metadata in the system
- Q access to O cannot be verified

- 10,862 objects' lifetime longer than 10s, has the problem
- average 22.87 times of modification during object lifetime
- solution: ML auditing





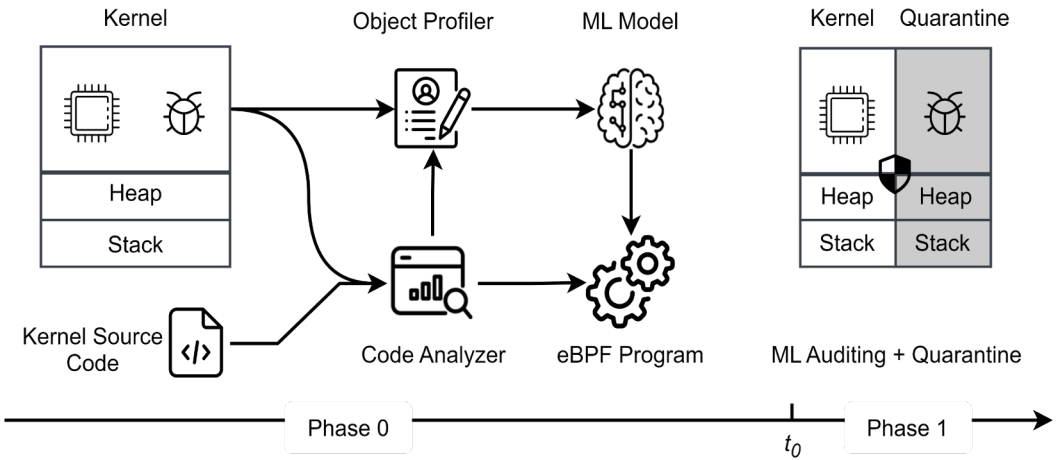
Security Model

- Kernel is trusted
- Untrusted components are confined within quarantine zone

	Kernel			Quarantine		
	read	**write**	**exec**	**read**	**write**	**exec**
Kernel Code	√		√	√		
Kernel Data	√	√		√		
Kernel Heap	√	√		√		
Kernel Stack	√	√		√		
O2Q	√	√	√	√		
Quarantine Code	√	√	√	√		√
Quarantine Data	√	√		√	√	
Quarantine Heap	√	√		√	√	
Quarantine Stack	√	√		√	√	

On-the-fly OS Quarantine: Workflow

- phase 0: build a sandbox, collect data, train model, synthesize eBPF program
- phase 1: load eBPF start protection



O2Q Phase 0: Code analyzer

Identify and enforce mechanisms related to mandatory execution directives, constraining data access and control flow within quarantine zones.

- Indirect jump instructions
- Memory write instructions
- Subject switch instructions

Performance optimization with 24.07% reduction in instrumenting

- Skip Determining Address
- Ignore stack access
- Ignore redundant checks
- Ignore return checks

```
Indirect jump: call *%rax
```

```
Memory write: mov $0x0, (%rsi, %rdx, 1)
```

```
Determined address: mov off(%rip), %rax
```

```
Stack frame create: sub offset, %rsp
```

```
Stack access: mov x, off (%rsp/rbp)
```

```
Redundant check: mov  $0x0, off1(%rsi)
```

```
Redundant check: mov  $0x0, off2(%rsi)
```

O2Q Phase 0: Model Training

- Object Profiler

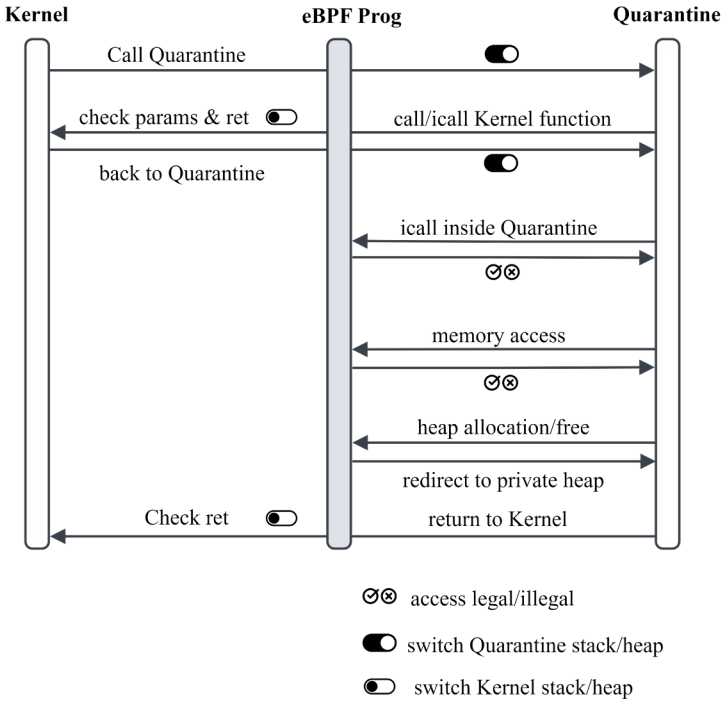
Feature	**Lable**
Data object content	Data object type/ if belong to quarantine
Collect when object released	Record stacktrace when allocate， Analyze object type offline

- Decision Tree Model

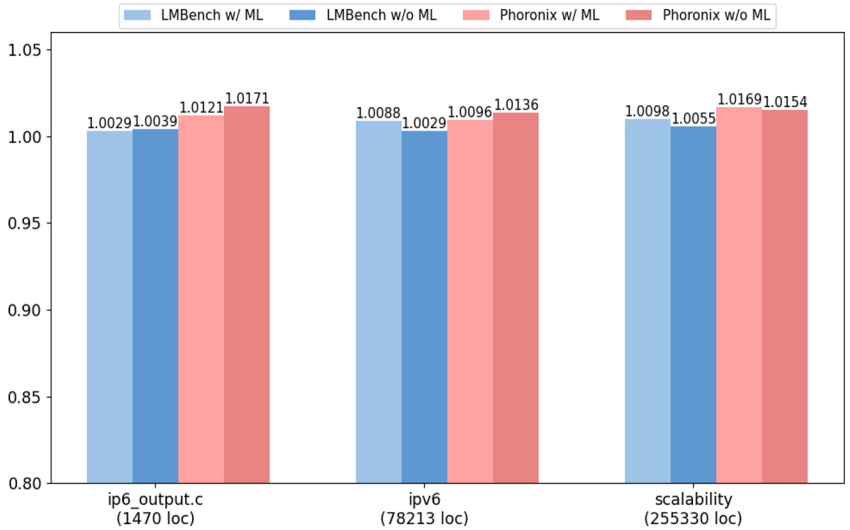
- Better suited for processing tabular data than deep learning
- Interpretable and have a defined execution time
- Does not lose quantitative accuracy of the model
- Can be converted to BPF implementation

O2Q: Phase1 Auditing and Quarantine

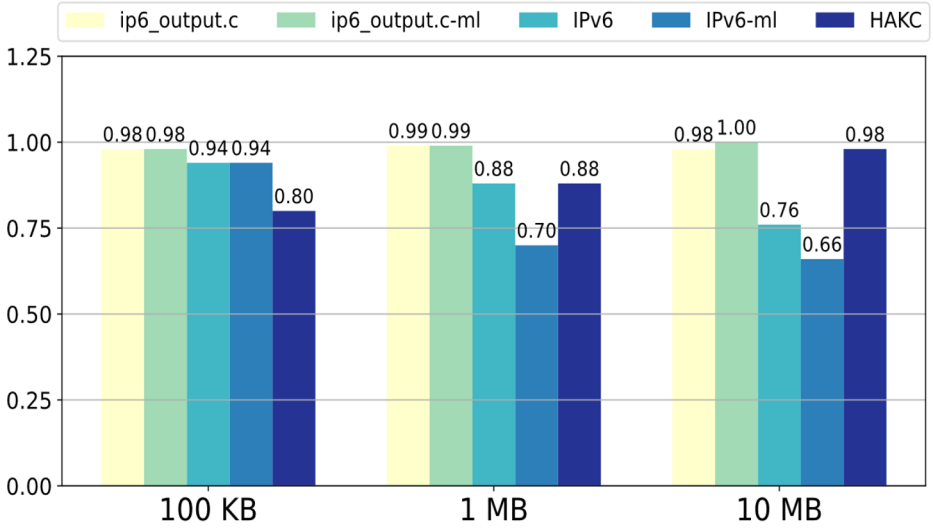
- Eliminate the complex vulnerabilities inside the Quaratine Zone
  - Control flow
  - Private heap & stack
  - Data object
  - Legitimate parameters and return values



O2Q: Evaluation



overhead to the system



performance loss to the quarantine zone

O2Q: Evaluation

	Object Type		Quarantine	
	Accuracy	**Macro F1**	**Accuracy**	**Macro F1**
IPV6				
Decision Tree	96.88 ± 0.65	75.56 ± 1.84	99.99 ±0.02	99.98 ± 0.03
Random Forest	96.91 ± 0.63	78.81 ± 0.73	100 ± 0.01	99.99 ± 0.01
Neural Network	89.63 ± 1.29	38.76± 2.70	99.99 ± 0.01	99.99 ±0.01
Sched				
Decision Tree	80.48 ± 0.76	71.04 ± 1.77	99.93 ± 0.14	97.74 ± 4.22
Random Forest	80.61 ± 0.69	76.28 ± 0.49	100 ± 0	99.99 ± 0.01
Neural Network	65.98 ± 6.91	39.18 ± 1.48	99.66±0.03	89.47±1.20
Netfilter				
Decision Tree	89.47 ± 0.23	78.17 ± 4.88	99.92 ± 0.07	99.51 ± 0.46
Random Forest	89.54 ± 0.15	81.87 ± 1.86	99.96 ± 0.05	99.77 ± 0.29
Neural Network	72.9 ± 2.23	37.98 ± 2.83	97.16 ±0.17	74 ± 2.56

performance of ML auditing

	Accuracy	**Macro F1**	**Accuracy**	**Macro F1**
Feature Length				
32	88.40 ± 0.42	73.97 ± 3.83	98.75 ± 0.41	91.91 ± 2.32
64	89.15 ± 0.33	77.24 ± 4.21	99.91 ± 0.07	99.47 ± 0.45
128	89.18 ± 0.29	77.44 ± 4.33	99.85 ± 0.1	99.46 ± 0.64
256	89.26 ± 0.29	77.34 ± 5.06	99.92 ± 0.08	99.51 ± 0.49
1024	89.47 ± 0.23	78.17 ± 4.88	99.92 ± 0.07	99.51 ± 0.46
Max Depth				
3	61.18 ± 2.45	1.72 ± 0.19	97.47 ± 0.4	79.34 ± 3.03
7	76.59 ± 2.38	8.48 ± 0.58	99.44 ± 0.21	96.44 ± 1.32
10	83.54 ± 2.19	21.06 ± 2.19	99.65 ± 0.14	97.78 ± 0.86
14	89.47 ± 0.23	78.17 ± 4.88	99.92 ± 0.07	99.51 ± 0.46

performance of tuning decision tree feature and depth

Thanks

Github Repo:
https://github.com/a8stract-lab/o2c