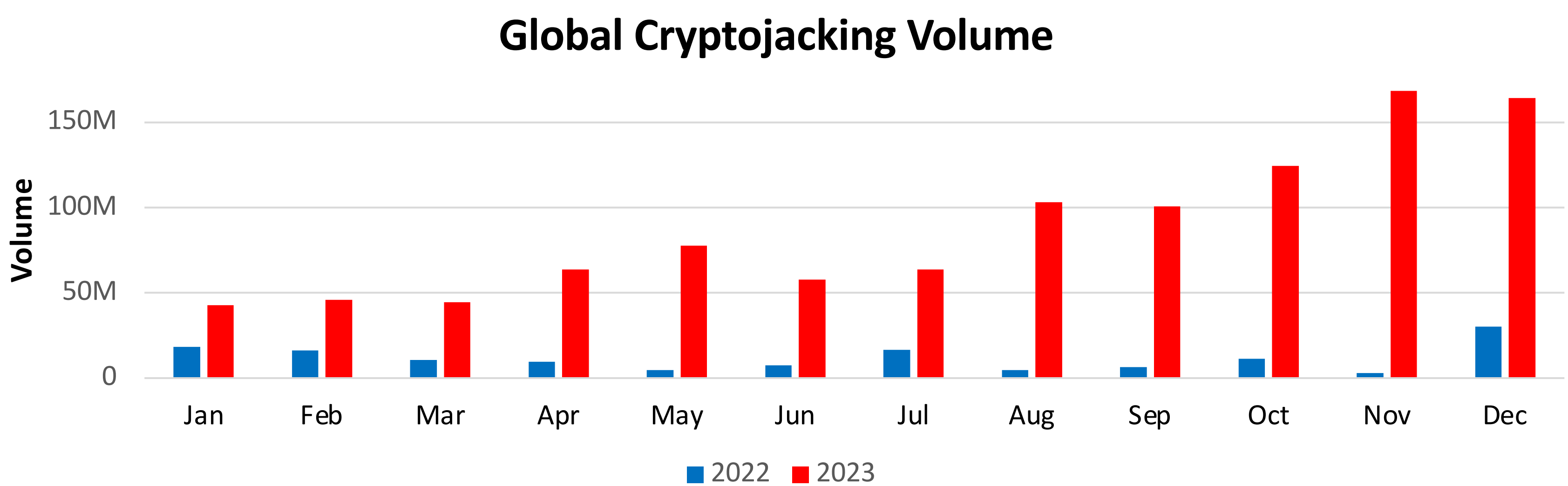
CryptoGuard: Lightweight Hybrid Detection and Response to Host-based Cryptojackers in Linux Cloud Environment

Gyeonghoon Park, Jaehan Kim, Jinu Choi, and Jinwoo Kim

The Growing Threat of Host-based Cryptojackers

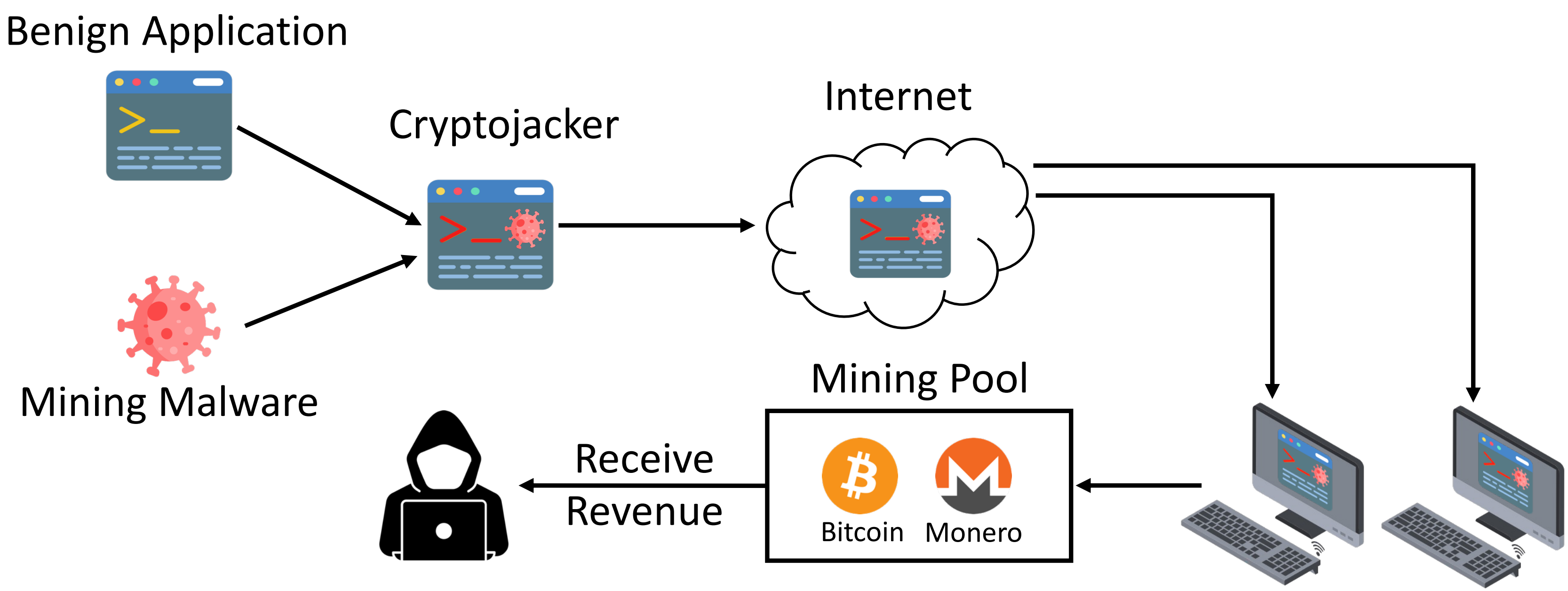
- Target Linux-based public cloud environments
  - In 2018, hackers enlisted Tesla's public cloud to mine cryptocurrency[1]
  - In 2023, leaked AWS credentials were used to create EC2 instances for cryptomining[2]
- Cryptojacking incidents have increased by 659%[3]



UCI Samueli School of Engineering University of California, Irvine

1 https://www.wired.com/story/cryptojacking-tesla-amazon-cloud/
2 https://thehackernews.com/2023/10/elektra-leak-cryptojacking-attacks.html
3 SonicWall, Cyber Threat Report 2024

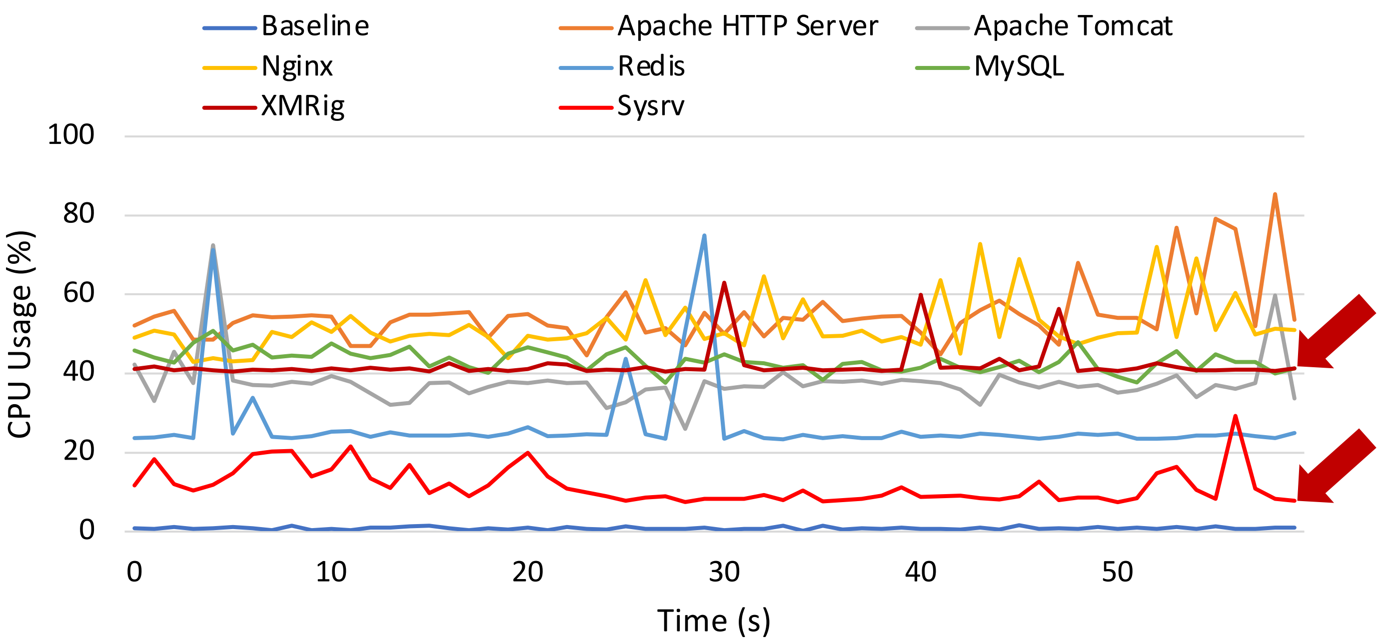
The Lifecycle of a Host-based Cryptojacker[4]



- ***Host-based Cryptojacking***
  - Exploits PCs and IoTs
  - Pervasive in modern cloud environments
- ***In-browser Cryptojacking***:
  - Exploits client web browsers connected to malicious websites
  - Slowed after CoinHive shutdown in 2019

Evasion Techniques of Host-based Cryptojackers

- **CPU throttling**
  – Makes difficult to determine cryptojackers using ***rule-based detection*** that relies on CPU usage



Evasion Techniques of Host-based Cryptojackers

- **PID obfuscation**
  - Continuously obfuscates PIDs with short and unexpected time intervals

PID	USER	CPU%	MEM%	TIME+	Command△
3879	user1	0.6	0.1	0:00.07	/cryptojacker.bin

- **Restoration via entry points**
  - Can restore cryptojackers even after a system reboot or process termination by compromising entry points (e.g., cronjob[5], rc.local[6])

```
GNU_nano 6.2                    /tmp/crontab.w8a6t5/crontab
* * * * * /home/user1/.cache/mesa_shader_cache/18/b4fqq09
```


5 https://www.crowdstrike.com/en-us/blog/new-kiss-a-dog-cryptojacking-campaign-targets-docker-and-kubernetes/
6 https://www.f5.com/labs/articles/threat-intelligence/attackers-use-new--sophisticated-ways-to-install-cryptominers

Existing Solutions

- **Non-ML detection**
  - Lachtar et al. [IEEE CAL ‘20], D. Tanana and G. Tanana [IEEE CAL ‘20]
- **ML detection**
  - Gomes et al. [NCA ‘20], Caprolu et al. [Comput. Commun. ‘21], Tekiner et al. [NDSS 2022]
  - Mani et al. [ACSOS ‘20]
- **Prevention solution**
  - Franco et al. [IEEE ICC ‘23] → Suricata IDS alerts

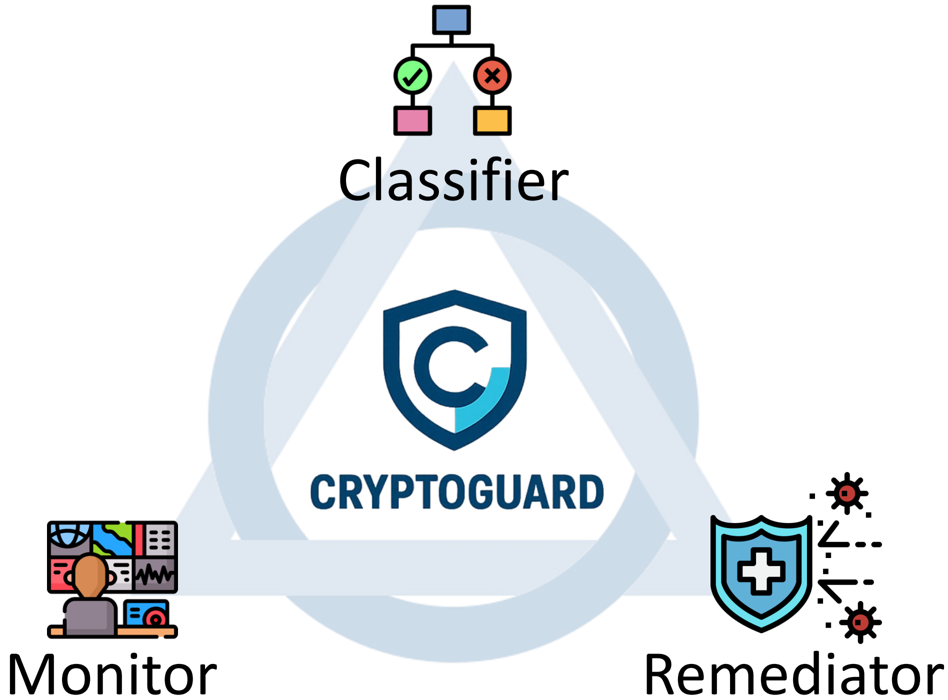
None of them focuses on both detection and persistent prevention system

Challenges

- How to minimize ***overhead*** when collecting fine-grained features for detection?
  - Network traffic, CPU, HPC, syscall, etc…
- How to counter ***evasion techniques*** in detection?
  - CPU throttling, PID obfuscation, entry point, etc…
- How to achieve ***scalability*** in the cloud environment?

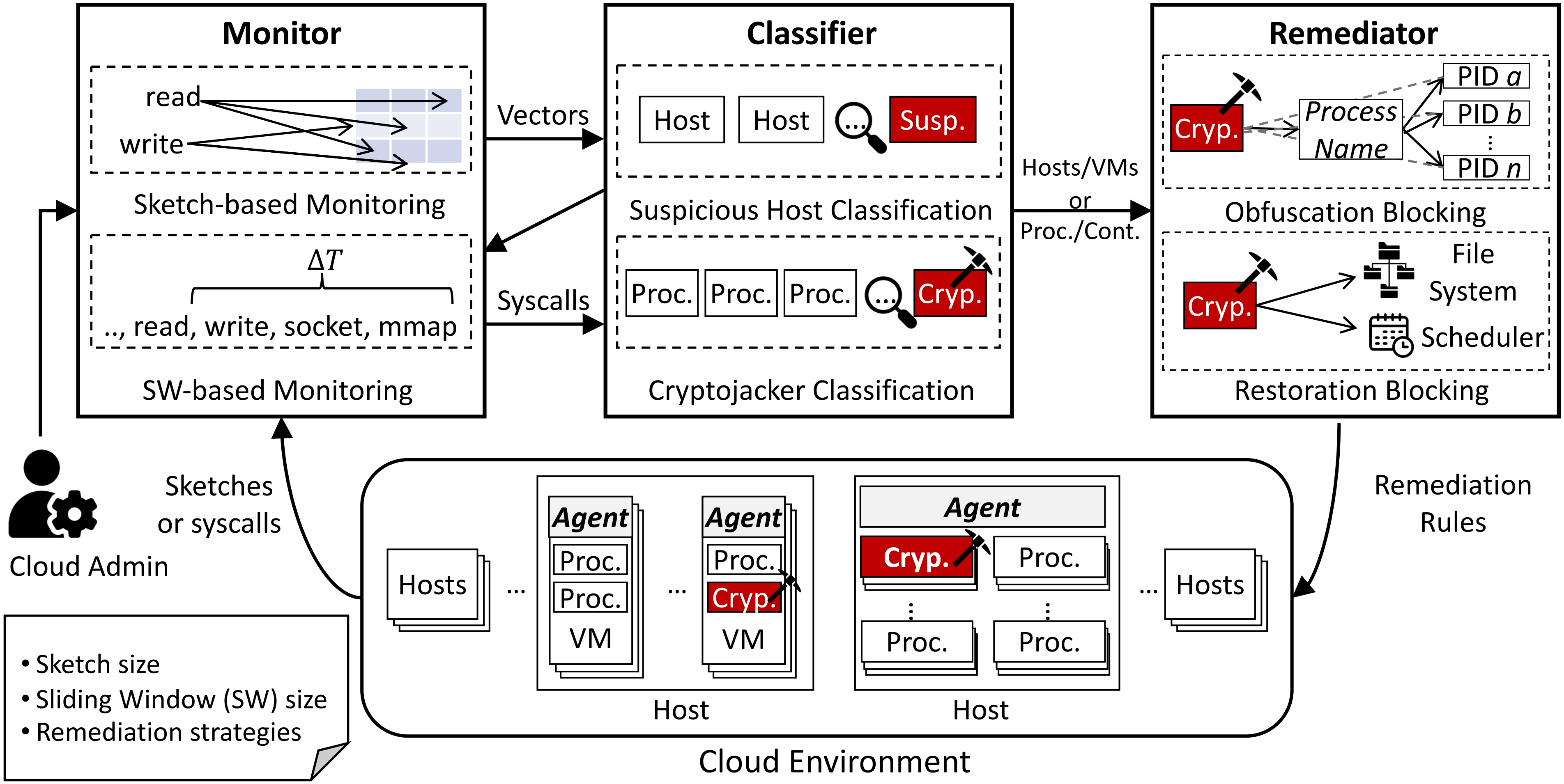
Cryptoguard

- Sketch/sliding window-based syscall monitoring via eBPF[7]
- Precise detection for stealthy cryptojackers via deep learning
- Integrated detection and remediation approaches



7 https://ebpf.io

CryptoGuard Overview

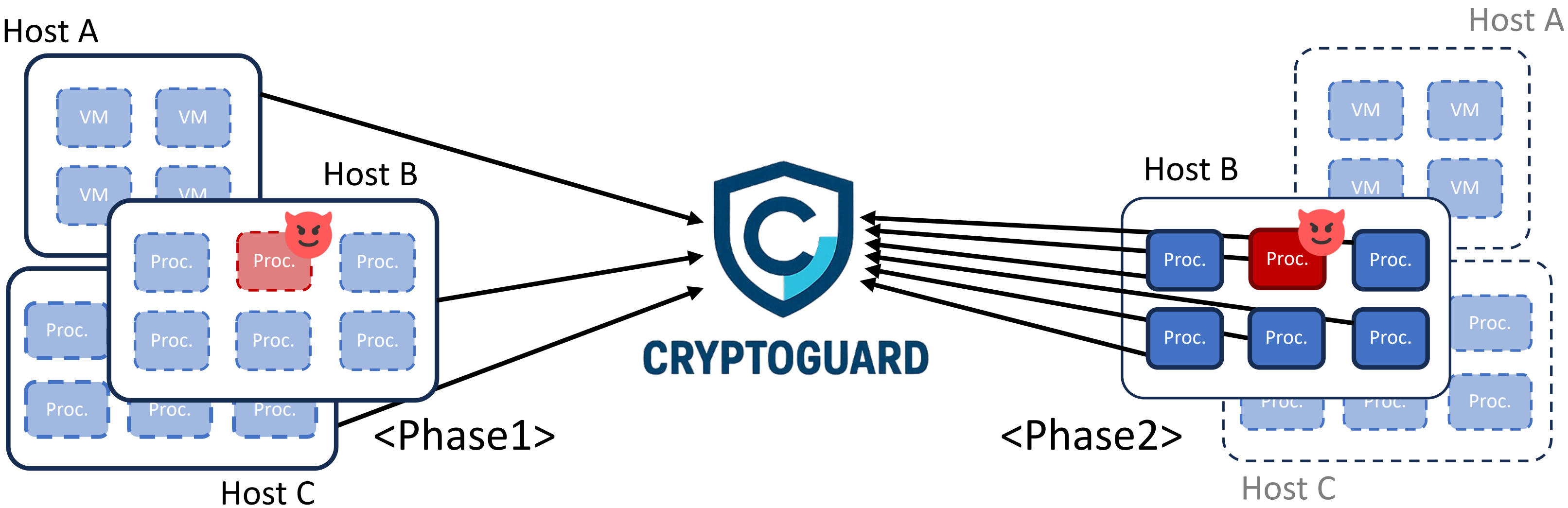


System Call Monitoring

- Q. Is it practical to monitor system calls in cloud environment?

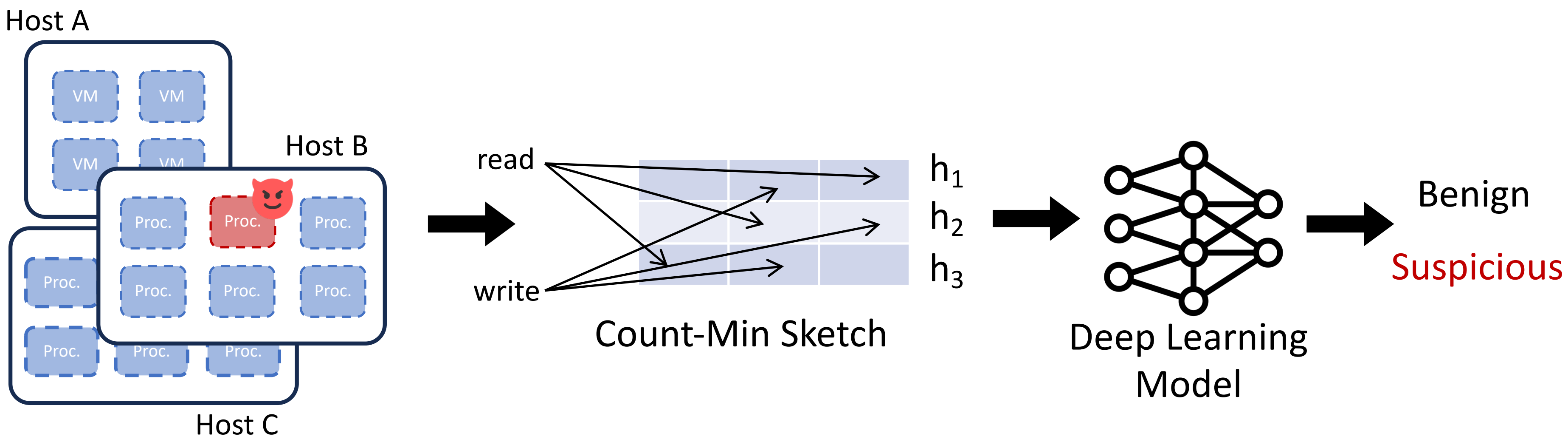
→ Monitoring all processes and containers imposes significant overhead

- Key idea: Perform monitoring in two phases
  - Phase 1: Host-level monitoring
  - Phase 2: Process-level monitoring



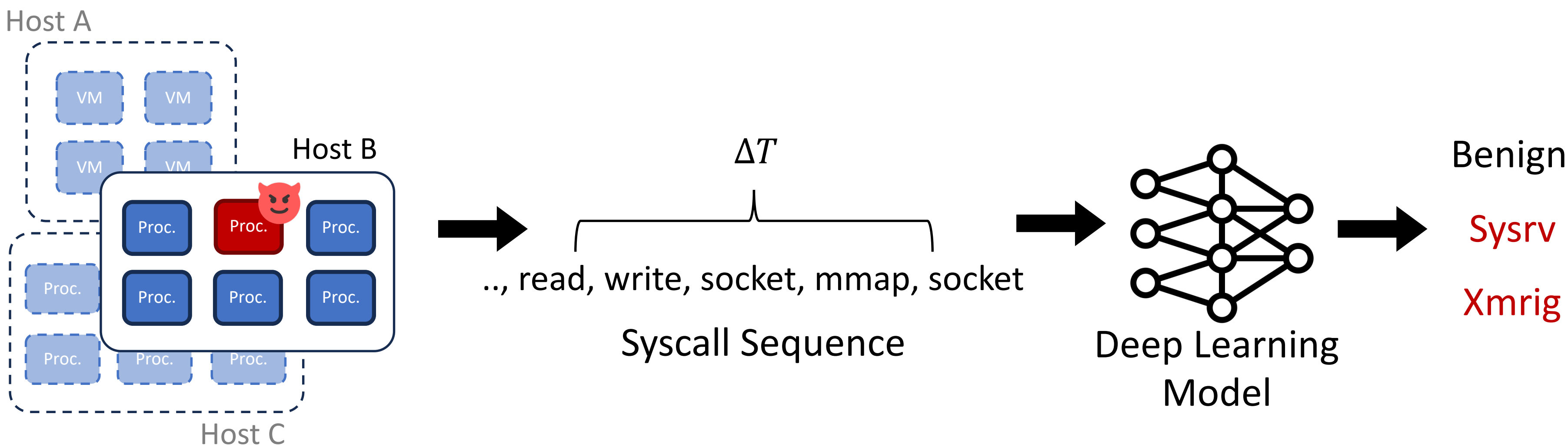
1st Phase: Host-level Monitoring/Detection

- Key idea: ***Count-Min Sketch (CMS)***[8]
  - A probabilistic data structure for frequency estimation
  - Record hashed values of system calls into a fixed-size 2D array
- Perform binary classification: suspicious/benign



2nd Phase: Process-level Monitoring/Detection

- Key idea: ***Sliding Window***
  - Trace syscalls within a time window of ΔT from the detection point
  - Recent traces are sufficient due to the distinctiveness of patterns
- Perform multi-class classification: xmrig/sysrv/benign



Cryptojacker Remediation

```
openat(AT_FDCWD, "/home/c
O_RDONLY|O_CLOEXEC) = 484
```
→ Check if the cache directory is a

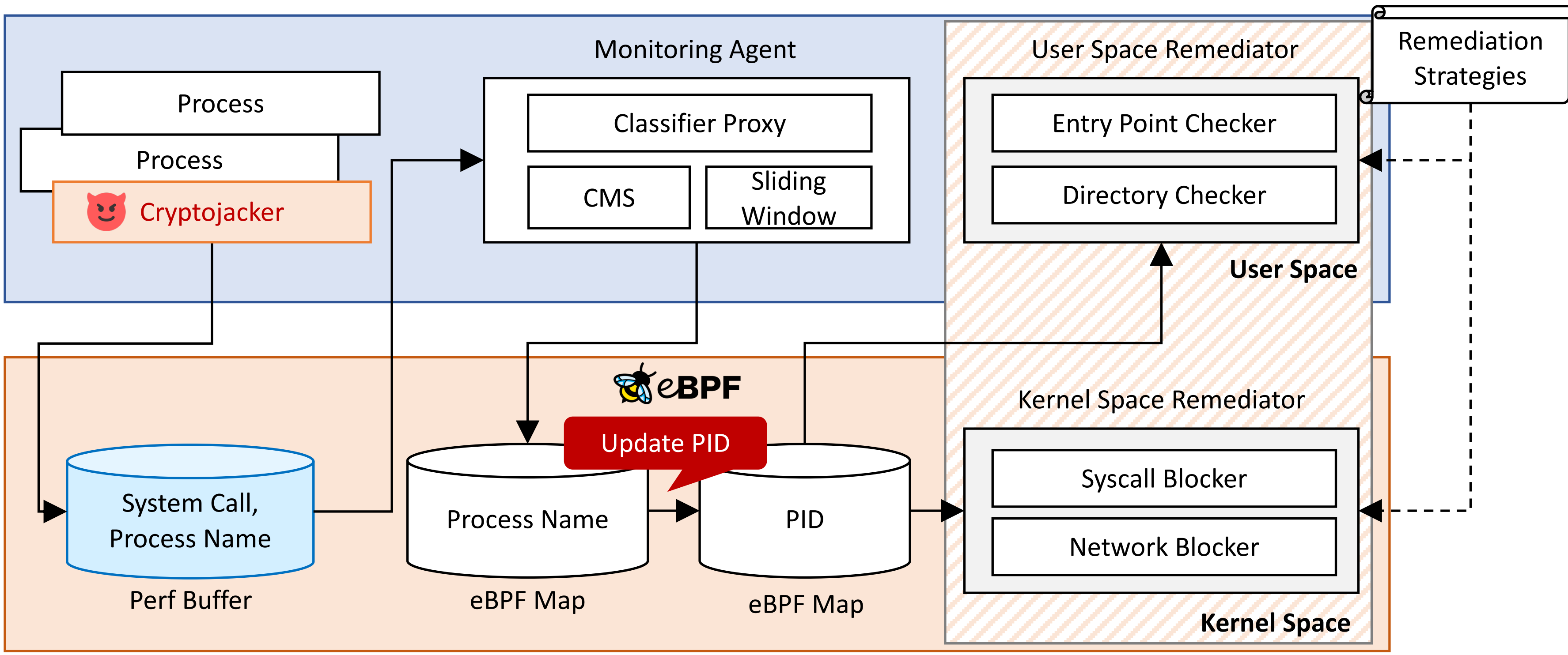
```
openat(AT_FDCWD, "/home/c
nvdmld", O_WRONLY|O_CREAT|O_TRUNC|O_CLOEXEC, 0777) = 484
```
→ Creates a new binary using a randomly generated alphanumeric string

Job Schedule_1 Before CryptoGuard:

```
* * * * * /home/cryptoguard/.cache/mesa_shader_cache/73/nvdmld
```

Deleted Job Schedule_1 After CryptoGuard:

~~* * * * * /home/cryptoguard/.cache/mesa_shader_cache/73/nvdmld~~



Dataset & Implementation

- Collected a dataset by executing both malware samples and benign processes
  - 123 real-world Linux cryptojacking malware samples[9] were used
  - Benign processes include Apache HTTP Server, Apache Tomcat, Nginx, Redis, MySQL

Family	# of Samples	Class
Sysrv	100	Sysrv
XMRig	15	XMRig
TeamTNT	5	
WatchDog	3	
Total	**123**	

- 5,000 lines of C code (libbpf and bpftrace)

9 VMware Threat Report 2022

Performance of Detecting Suspicious Hosts (1st Phase)

- Binary classification to distinguish suspicious hosts from benign ones
- LSTM and CNN classifiers achieved average F1-scores of 96.42% and 95.82%

Sketch Size	Class	LSTM			CNN		
		Precision	Recall	F1-score	Precision	Recall	F1-score
272×3	Benign	94.62%	98.88%	96.70%	97.36%	94.39%	95.85%
	Malware	98.74%	94.00%	96.31%	94.18%	97.25%	95.69%
55×3	Benign	95.82%	96.21%	96.51%	95.43%	95.28%	95.36%
	Malware	95.84%	96.51%	96.71%	94.75%	94.91%	94.83%
55×5	Benign	95.67%	97.63%	96.64%	94.77%	**98.88%**	**96.78%**
	Malware	97.32%	95.11%	96.20%	**98.74%**	94.17%	96.40%

Performance of Detecting Cryptojacker Processes (2nd Phase)

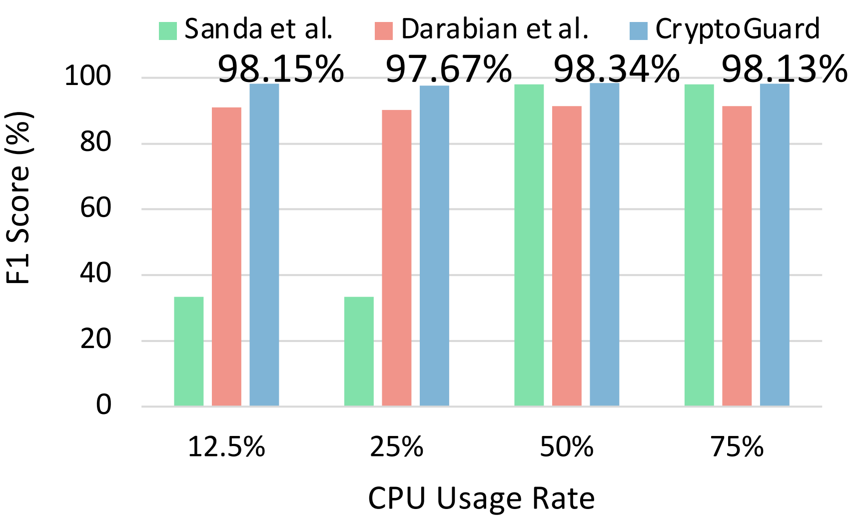
- For ΔT= 30, the CNN classifier achieved F1-scores of 95.62%, 98.54%, and 98.87%

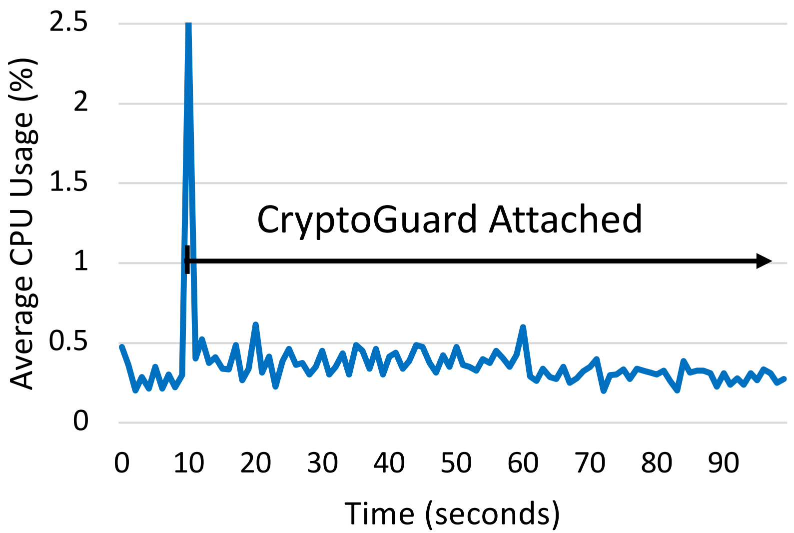
Sliding Window Size	Class	LSTM			CNN		
		Precision	Recall	F1-score	Precision	Recall	F1-score
$\Delta T = 30$	XMRig	75.00%	92.14%	82.69%	97.76%	93.57%	95.62%
	Sysrv	91.50%	90.91%	91.21%	98.70%	98.38%	98.54%
	Benign	98.10%	93.66%	95.83%	98.36%	**99.40%**	**98.87%**
$\Delta T = 60$	XMRig	69.11%	87.63%	77.27%	91.40%	87.63%	89.47%
	Sysrv	90.85%	89.68%	90.26%	**99.34%**	96.77%	98.04%
	Benign	95.76%	89.68%	92.62%	96.89%	98.94%	97.91%
$\Delta T = 90$	XMRig	69.11%	87.63%	77.27%	91.40%	87.63%	89.47%
	Sysrv	90.85%	89.68%	90.26%	**99.34%**	96.77%	98.04%
	Benign	95.76%	89.68%	92.62%	96.89%	98.94%	97.91%

UCI Samueli School of Engineering
University of California, Irvine

Evasion Resilience / Overhead

- F1-scores of the three approaches under different CPU usage rates of 12.5%, 25%, 50%, and 75%
  - CryptoGuard offers resilience to CPU throttling evasion attacks
- CPU usage variation during monitoring
  - CryptoGuard imposes minimal overhead (0.29% → 0.35%)





Conclusion

- Host-based Cryptojacker
  - Stealthy behavior
  - Employs obfuscation techniques
  - The vast number of hosts and processes in cloud environments makes detection difficult

- **CryptoGuard**: a lightweight solution for detecting and remediating host-based cryptojacking in cloud environment
  - Sketch-based and sliding window-based monitoring
  - Counters the persistence mechanism by Cryptojacker
  - https://github.com/PGHOON/CryptoGuard

Thank you for listening

(hoonp2@uci.edu)

ACKNOWLEDGMENTS: This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00457937, Design and implementation of security layers for secure WebAssembly-based serverless environments, 50%) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00118, RS-2021-II210118, Development of decentralized consensus composition technology for large-scale nodes, 50%).