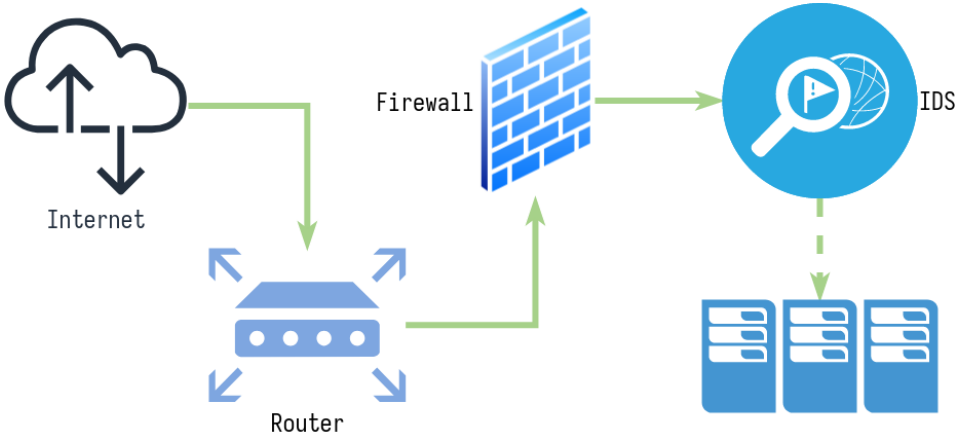
An Empirical Analysis of IDS Approaches in Container Security

Yigit Sever, Goktug Ekinci, Adnan Dogan, Bugra Alparslan, Said Gurbuz, Vahab Jabrayilov, Pelin Angin

Middle East Technical University

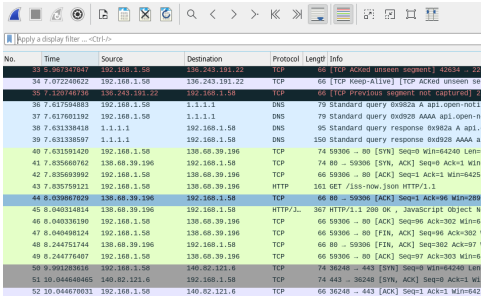
Intrusion Detection Systems



IDS Approaches

Network-based IDS

- sensor on data “on the wire”

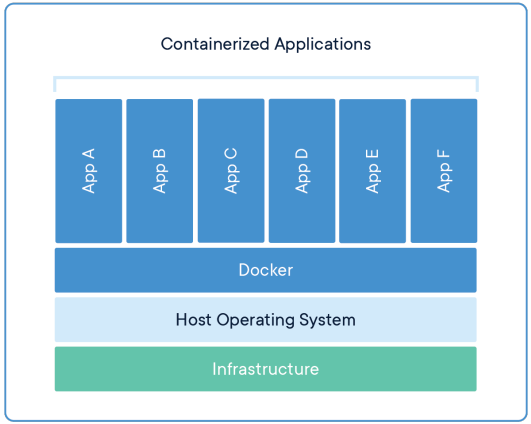


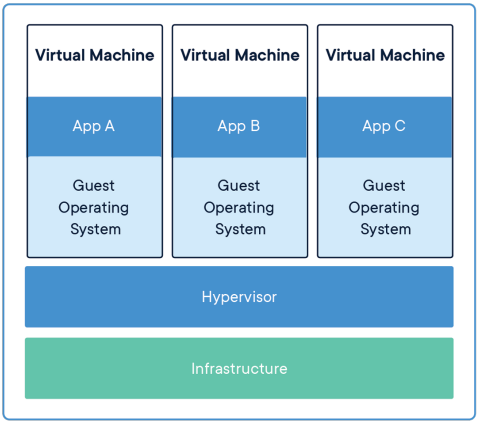
Host-based IDS

- sensor on machine behaviour

```
50067 22:20:56.305073049 12 sudo (3494636.3494636) > rt_sigaction
50068 22:20:56.305073141 12 sudo (3494636.3494636) < rt_sigaction
50069 22:20:56.305073274 14 <NA> (<NA>.0) > switch next=3492401 pgft_maj=0 pgft_min=0 vm_siz
50070 22:20:56.305073335 12 sudo (3494636.3494636) > rt_sigaction
50072 22:20:56.305073426 12 sudo (3494636.3494636) < rt_sigaction
50073 22:20:56.305073517 12 sudo (3494636.3494636) > read fd=9(<f>/dev/ptmx) size=65536
50075 22:20:56.305074865 12 sudo (3494636.3494636) < read res=191 data=2530 22:20:56.2982426
0m.3494132) < .[0
50076 22:20:56.305074238 12 sudo (3494636.3494636) > rt_sigaction
50077 22:20:56.305074314 12 sudo (3494636.3494636) < rt_sigaction
50078 22:20:56.305074448 12 sudo (3494636.3494636) > rt_sigprocmask
50079 22:20:56.305074552 12 sudo (3494636.3494636) < rt_sigprocmask
50080 22:20:56.305074626 7 <NA> (<NA>.0) > switch next=3420972 pgft_maj=0 pgft_min=0 vm_size
50081 22:20:56.305074642 14 <NA> (<NA>.3492401) > switch next=0 pgft_maj=0 pgft_min=0 vm_siz
50082 22:20:56.305074649 12 sudo (3494636.3494636) > rt_sigprocmask
50083 22:20:56.305074722 12 sudo (3494636.3494636) < rt_sigprocmask
50084 22:20:56.305074928 12 sudo (3494636.3494636) > ppoll fds=11:u1 3:p1 8:f4 8:f1 9:f1 tim
50085 22:20:56.305076014 0 sshd (3494132.3494132) < brk res=5592B03E6000 vm_size=17764 vm_rs
50086 22:20:56.305075085 7 <NA> (<NA>.3420972) > switch next=0 pgft_maj=0 pgft_min=0 vm_size
50088 22:20:56.305075455 12 sudo (3494636.3494636) < ppoll res=2 fds=8:f4 9:f1
50089 22:20:56.305075662 0 sshd (3494132.3494132) > read fd=10(<f>/dev/ptmx) size=32768
50090 22:20:56.305075694 12 sudo (3494636.3494636) > rt_sigaction
50091 22:20:56.305075782 12 sudo (3494636.3494636) < rt_sigaction
50092 22:20:56.305075873 12 sudo (3494636.3494636) > write fd=8(<f>/dev/tty) size=191
50094 22:20:56.305076665 12 sudo (3494636.3494636) < write res=191 data=2530 22:20:56.298242
00m.3494132) < .[0
50095 22:20:56.305076854 12 sudo (3494636.3494636) > rt_sigaction
50096 22:20:56.305076931 12 sudo (3494636.3494636) < rt_sigaction
50097 22:20:56.305076994 0 sshd (3494132.3494132) < read res=728 data=2523 22:20:56.29824166
0m.3494636) < .[
50098 22:20:56.305077078 7 <NA> (<NA>.0) > switch next=3420972 pgft_maj=0 pgft_min=0 vm_size
50099 22:20:56.305077124 12 sudo (3494636.3494636) > rt_sigaction
50101 22:20:56.305077211 12 sudo (3494636.3494636) < rt_sigaction
50102 22:20:56.305077304 12 sudo (3494636.3494636) > read fd=9(<f>/dev/ptmx) size=65536
50103 22:20:56.305077545 0 sshd (3494132.3494132) > rt_sigprocmask
50104 22:20:56.305077654 0 sshd (3494132.3494132) < rt_sigprocmask
```

Containers vs. Virtual Machines





Existing IDS Literature?

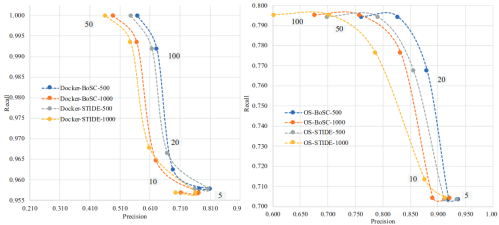


Fig. 10. Docker and OS Recall-Precision curves for epoch size and detection threshold impact analysis.

“The experiments in the OS deployment led to worse results than for Docker and LXC, indicating that besides the practical advantages, there is also an added effectiveness due to a more precise definition of the monitoring surface possible in the containers …” [FGA20]

Progress

Microservices

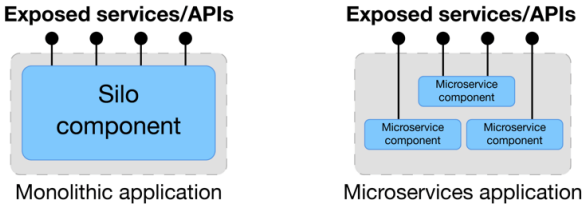


Figure from Microservices architecture: IBM's POV

Microservices architecture uses software containerization

Literature Survey

Author	Monitor
Srinivasan et al. [Sri+18]	`strace`
Abed et al. [ACL15]	`strace`
Cavalcanti et al. [CIF21]	Sysdig
Flora et al. [FGA20]	Sysdig
Tien et al. [Tie+19]	Sysdig
Tunde-Onadele et al. [Tun+19]	Sysdig
Röhling et al. [Röh+19]	Sysdig

Another Look at the Literature Survey

Author	**Application**	**Benign**	**Malicious**
Srinivasan et al.	DVWA	*unknown*	`sqlmap`
Abed et al.	MySQL	`mysqlslap`	`sqlmap`
Cavalcanti et al.	MySQL	TPC-C	TPC-C
Flora et al.	MariaDB	TPC-C	exploit-db
Röhling et al.	MariaDB	*various tools*	*various tools*

Progress

Recap

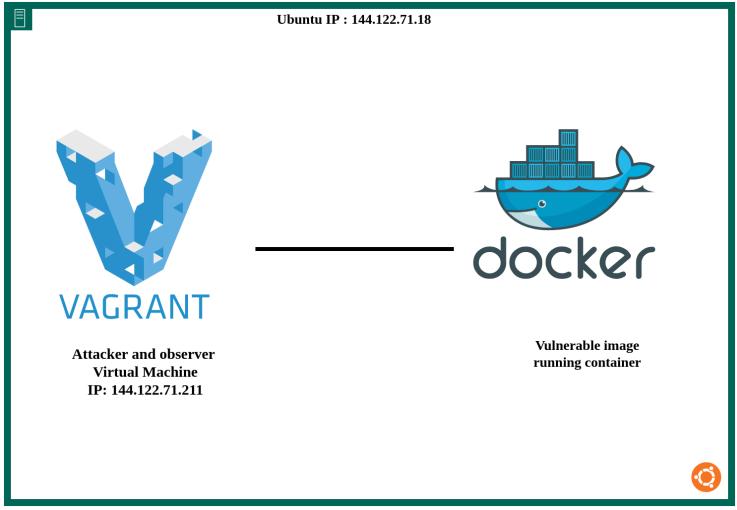
- The literature is using `syscalls` for container IDS
- Targets are *mostly* limited to databases and old workloads

Objectives

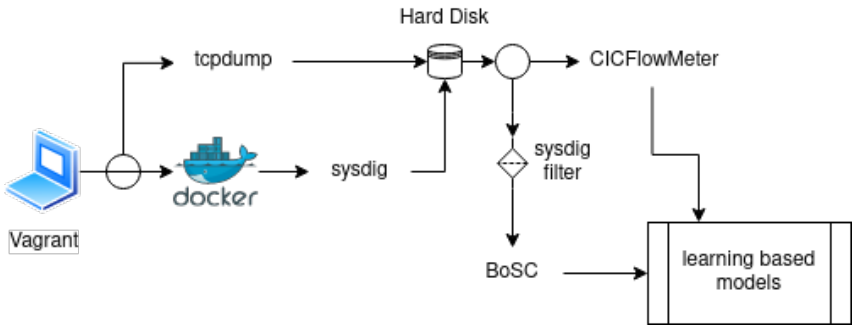
- Compare `syscall` monitoring with network based monitoring
- Do it using recent applications and attacks

Progress

Environment Setup



Experiment Pipeline



CWEs and CVEs

- CWE: Nature of the vulnerability
- CVE: Vulnerability in action

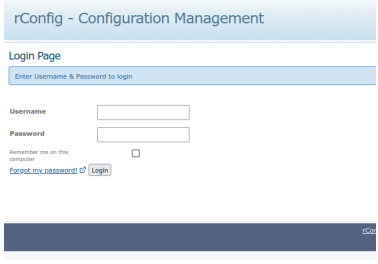
CVE	CWE
CVE-2019-16662	CWE-78
CVE-2019-19509	CWE-78
CVE-2020-10220	CWE-89

- CWE-78: OS Command Injection
- CWE-89: SQL Command Injection

Considerations

- Use well-known CVEs and attack tools
- Craft realistic user traffic, make sure that the generated user traffic does not follow any statistical distribution
- Do not consider environment-specific variables as features, e.g. IP addresses [VSO17; Sha+17].

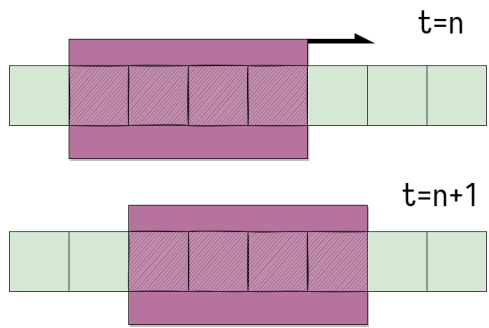
Vulnerable Application



rConfig
NETWORK MANAGEMENT

Bag-of-System-Calls (BoSC)

- Frequency of `syscalls` in a period of time
- Input is list of `syscalls` during attacks and regular user traffic
- At every timestep, we get a fixed vector where the `syscall` at position *n* has been encountered *x* times
- 332 `syscalls` in total



Network Flow

- Feature extraction method is updated fork of CICFlowMeter [ERJ21]
- Input is network packets from `tcpdump` during attacks and regular user traffic
- Yields features such as Packet Length Variance, Average Segment Size ...

Progress

Recap

We generated a dataset with malicious and benign traffic

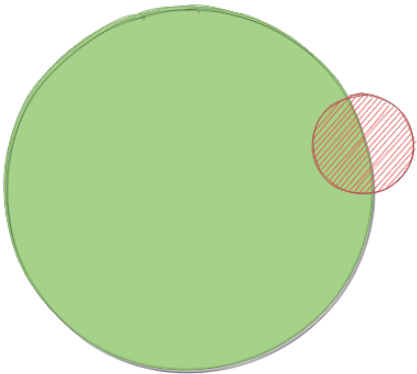
Compared `syscall` monitoring & network flow monitoring

Size of the Dataset

- The network flow dataset includes 279340 benign flows and 4532 malicious flows

- The BoSC dataset includes 4965 benign BoSC vectors and 134 malicious BoSC vectors

Base Rate Fallacy



Marking everything as benign would give 98.4% and 97.4% accuracy respectively!

Results - Network Flow

Model	TP	FP	Precision	Recall	Label
REPTree	1.000	0.002	1.000	1.000	B
	0.998	0.000	0.999	0.998	M
R. Tree	1.000	0.003	1.000	1.000	B
	0.997	0.000	0.999	0.997	M
R. Forest	1.000	0.002	1.000	1.000	B
	0.998	0.000	1.000	0.998	M
SMO	1.000	0.013	1.000	1.000	B
	0.987	0.000	0.998	0.987	M

Results - BoSC

Model	TP	FP	Precision	Recall	Label
REPTree	0.998	0.007	1.000	0.998	B
	0.993	0.002	0.937	0.993	M
R. Tree	0.998	0.030	0.999	0.998	B
	0.970	0.002	0.942	0.970	M
R. Forest	0.999	0.007	1.000	0.999	B
	0.993	0.001	0.964	0.993	M
SMO	0.998	0.000	1.000	0.998	B
	1.000	0.002	0.944	1.000	M

Comparison

BoSC

	a	b	Actual
REPTree	49561	9 133	a = 0 b = 1
R. Tree	49574	8 130	a = 0 b = 1
R. Forest	49601	5 133	a = 0 b = 1
SMO	49570	8 134	a = 0 b = 1

Network Flow

	a	b	Actual
REPTree	2793368	4 4524	a = 0 b = 1
R. Tree	27933213	8 4519	a = 0 b = 1
R. Forest	2793389	2 4523	a = 0 b = 1
SMO	27933360	7 4472	a = 0 b = 1

Conclusion & Future Work

- Network flow performed better across the board

- More attacks with better variation
- More applications
- Anomaly detection rather than classification

Questions?

Yigit Sever
yigitsever.com

Thank you :)

References I

[FGA20] José Flora et al. “Using Attack Injection to Evaluate Intrusion Detection Effectiveness in Container-based Systems”. In: *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*. Dec. 2020, pp. 60–69.

[Sri+18] Siddharth Srinivasan et al. “Probabilistic Real-Time Intrusion Detection System for Docker Containers”. In: *SSCC*. 2018.

[ACL15] A. S. Abed et al. “Intrusion Detection System for Applications Using Linux Containers”. In: *STM* (2015).

References II

[CIF21] Marcos Cavalcanti et al. “Performance Evaluation of Container-Level Anomaly-Based Intrusion Detection Systems for Multi-Tenant Applications Using Machine Learning Algorithms”. In: *The 16th International Conference on Availability, Reliability and Security*. ARES 2021. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 1–9. ISBN: 978-1-4503-9051-4.

[Tie+19] Chin-Wei Tien et al. “KubAnomaly: Anomaly Detection for the Docker Orchestration Platform with Neural Network Approaches”. In: *Engineering Reports* 1.5 (2019), e12080. ISSN: 2577-8196.

References III

[Tun+19] Olufogorehan Tunde-Onadele et al. “A Study on Container Vulnerability Exploit Detection”. In: *2019 IEEE International Conference on Cloud Engineering (IC2E)*. June 2019, pp. 121–127.

[Röh+19] Martin Max Röhling et al. “Standardized Container Virtualization Approach for Collecting Host Intrusion Detection Data”. In: *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. Sept. 2019, pp. 459–463.

[VSO17] Eduardo K. Viegas et al. “Toward a Reliable Anomaly-Based Intrusion Detection in Real-World Environments”. In: *Computer Networks* 127 (Nov. 2017), pp. 200–216. ISSN: 1389-1286.

References IV

[Sha+17] Iman Sharafaldin et al. “Towards a Reliable Intrusion Detection Benchmark Dataset”. In: *Software Networking* 2017.1 (2017), pp. 177–200. ISSN: 2445-9739.

[ERJ21] Gints Engelen et al. “Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study”. In: *2021 IEEE Security and Privacy Workshops (SPW)*. San Francisco, CA, USA: IEEE, May 2021, pp. 7–12. ISBN: 978-1-66543-732-5.