Bangalore, June 4th 2026

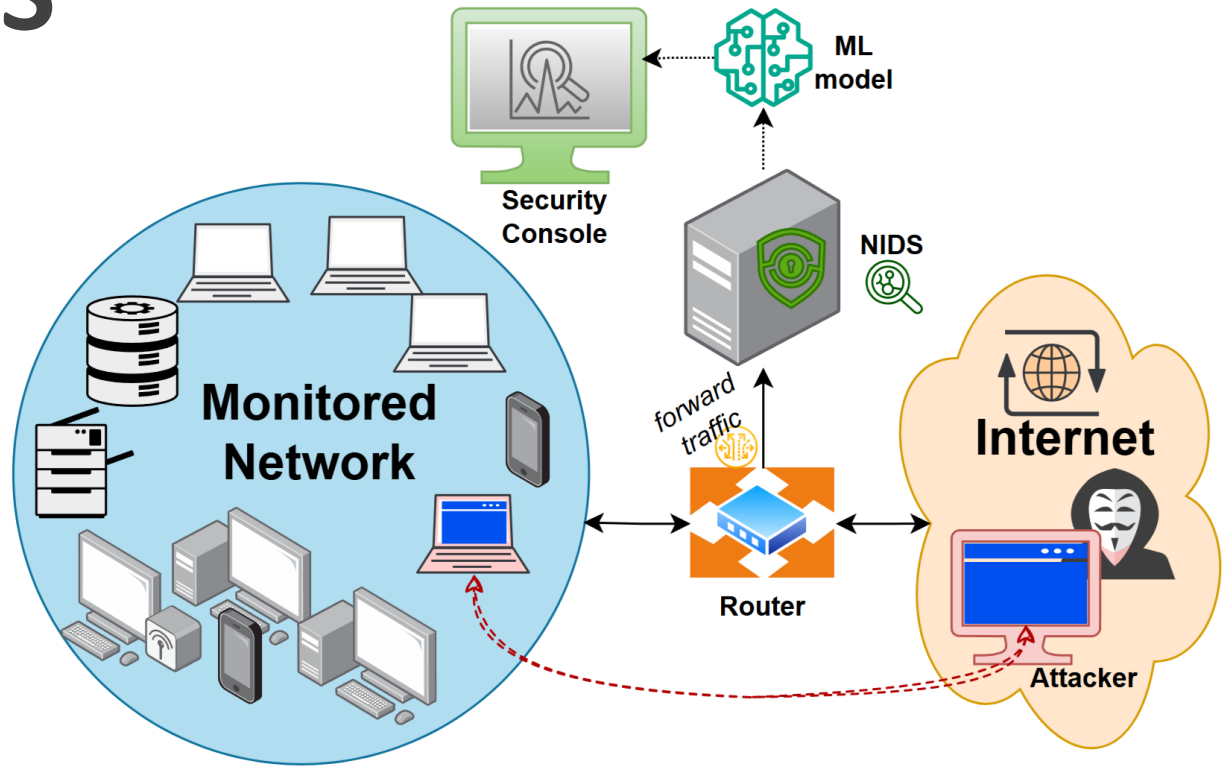
21st ACM ASIA Conference on Computer and Communications Security

“What is the Problem Space?” Defining Host-space Adversarial Perturbations against Network Intrusion Detection Systems

Miel Verkerken, Laurens d’Hooge, Bruno Volckaert, Filip de Turck, Giovanni Apruzzese

HÁSKÓLINN Í REYKJAVÍK | REYKJAVIK UNIVERSITY

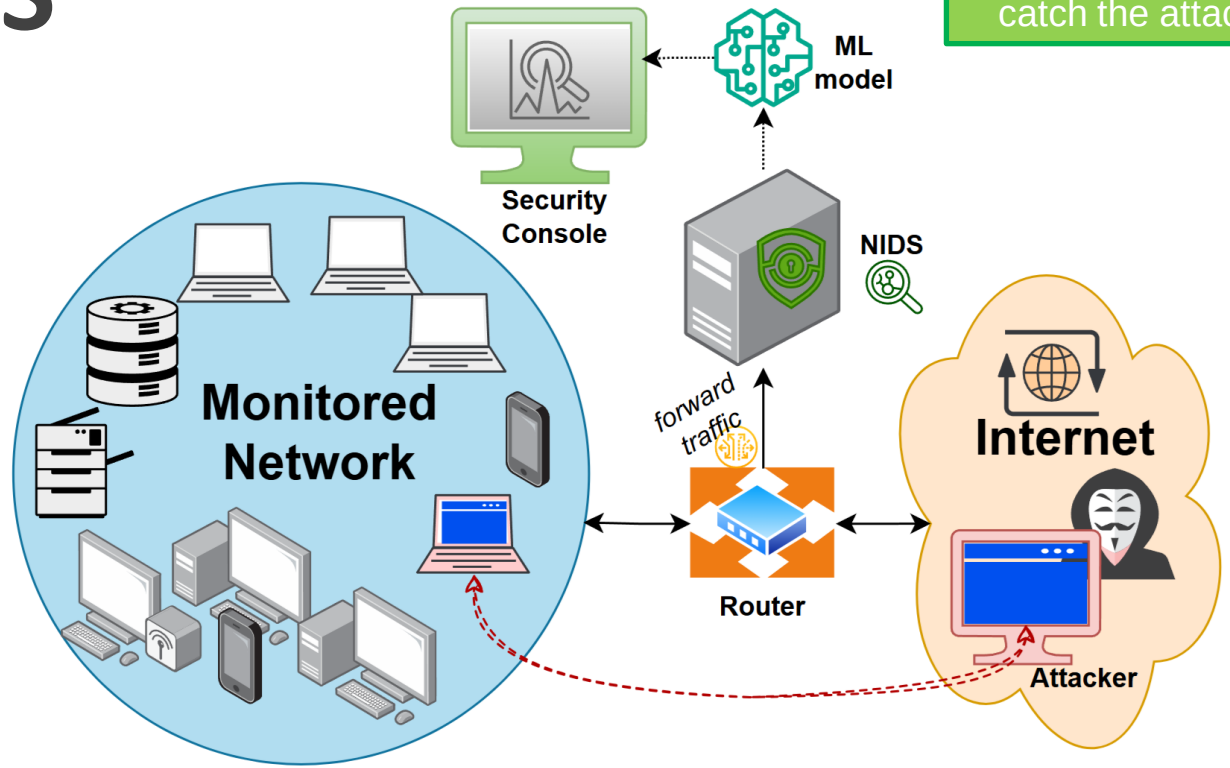
NIDS



Giovanni Apruzzese
giovannia@ru.is

NIDS

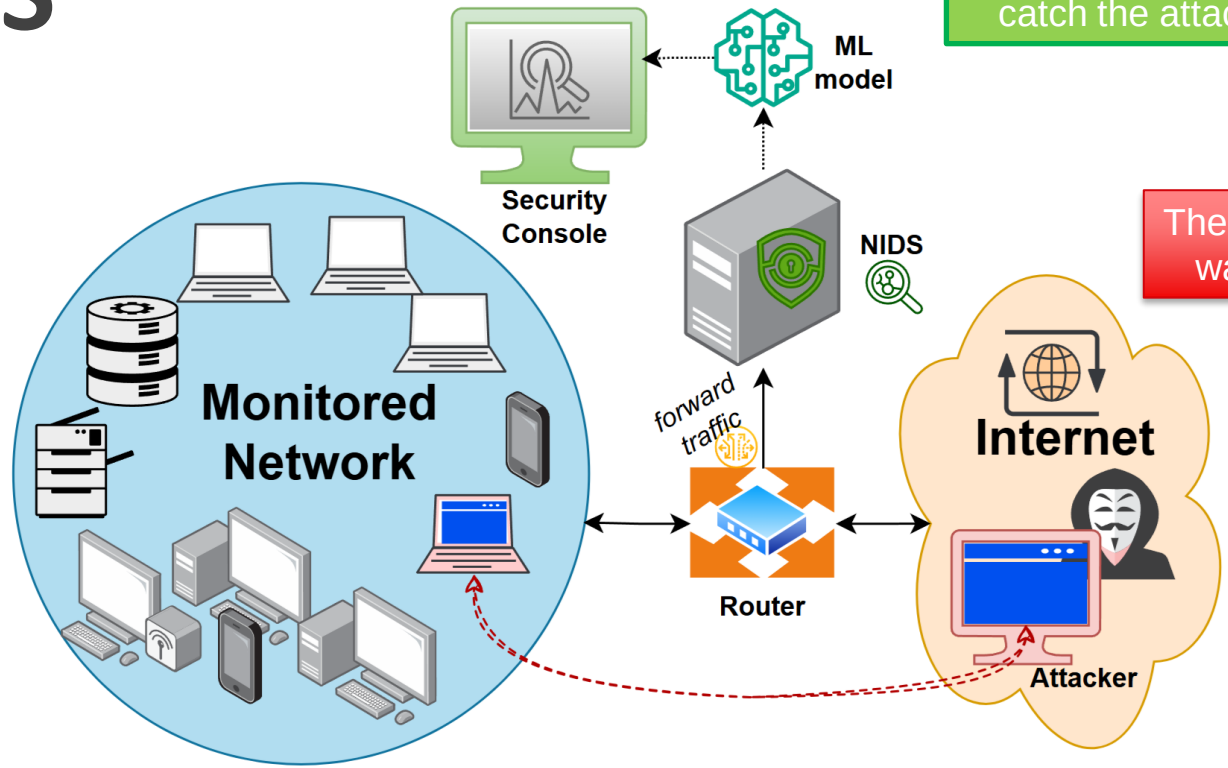
The NIDS wants to catch the attacker



Giovanni Apruzzese
giovannia@ru.is

NIDS

The NIDS wants to catch the attacker

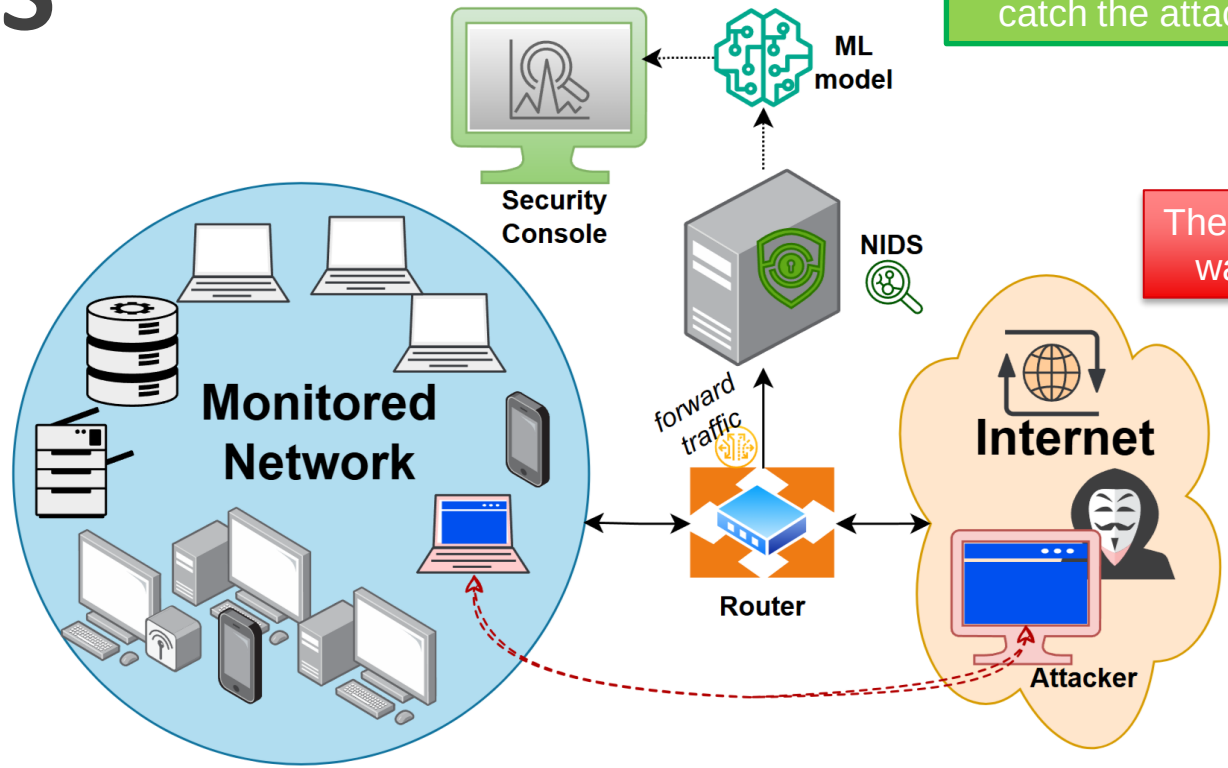


The attacker does not want to be caught

Giovanni Apruzzese
giovannia@ru.is

NIDS

The NIDS wants to catch the attacker

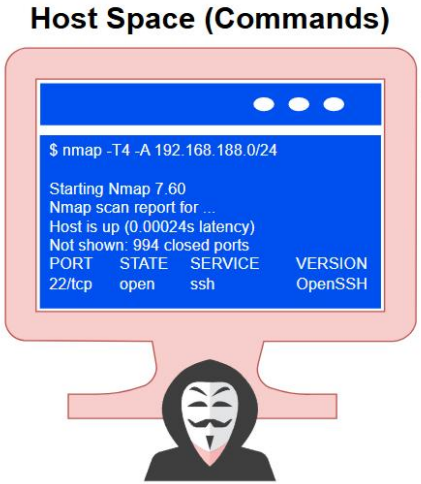


The attacker does not want to be caught

What is the attacker's problem space?

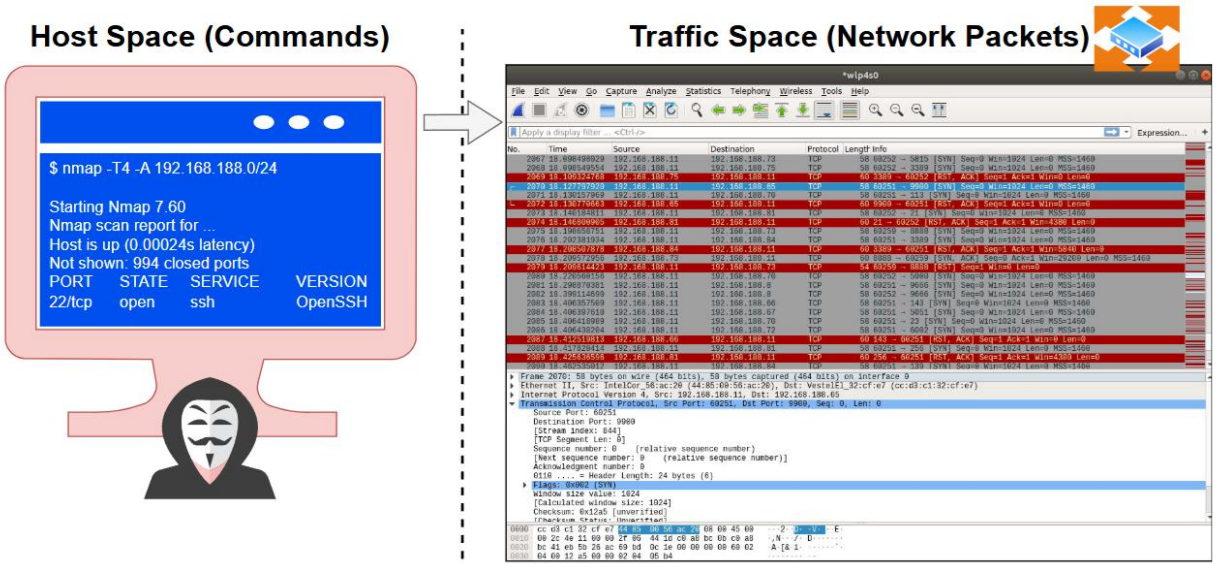
Giovanni Apruzzese
giovannia@ru.is

Example: PortScan



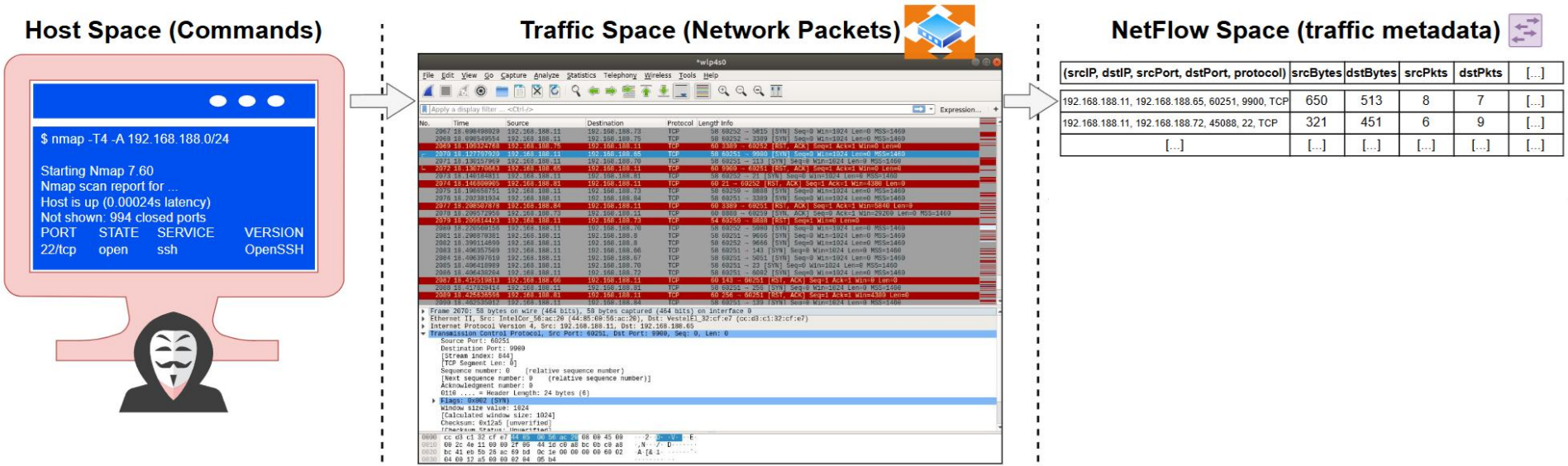
Giovanni Apruzzese
giovannia@ru.is

Example: PortScan



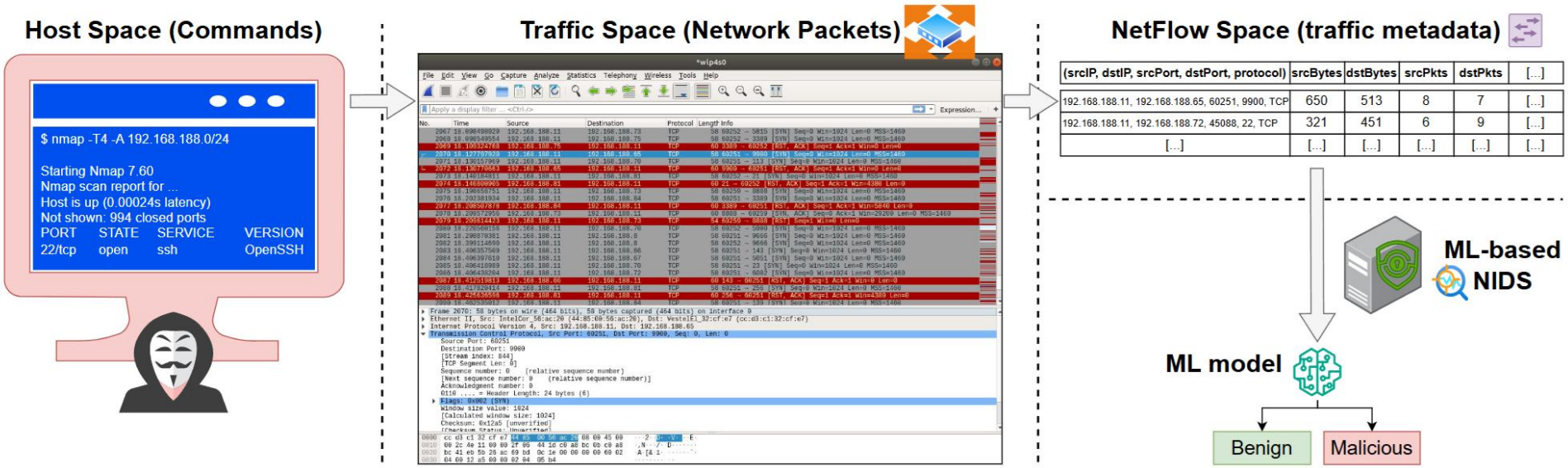
Giovanni Apruzzese
giovannia@ru.is

Example: PortScan



Giovanni Apruzzese
giovannia@ru.is

Example: PortScan



Giovanni Apruzzese
giovannia@ru.is

Takeaway

Real attackers perform *actions* leading to changes in the network data generated by their controlled hosts.

Giovanni Apruzzese
giovannia@ru.is

Takeaway

Real attackers perform *actions* leading to changes in the network data generated by their controlled hosts.

From an adversarial ML viewpoint, this can be simulated through “host-space perturbations” (HsP), which require operating at the host level—and not on the data that, despite having been generated by such ahost, was captured by an appliance outside the attacker’s control.

Giovanni Apruzzese
giovannia@ru.is

Threat Model

Given an attacker having a goal $\mathcal{G}$, which they pursue by using their knowledge $\mathcal{K}$ and capabilities $\mathcal{C}$ on the targeted system to devise a certain strategy $\mathcal{S}$.
To realise a valid HsP, the following must be met:

Giovanni Apruzzese
giovannia@ru.is

Threat Model – Goal

The overarching goal is the same as in the original threat model (i.e., $\mathcal{G}$), but it must also include evasion of an (ML-based) NIDS.

Otherwise, the attacker would not even attempt any evasion!

Giovanni Apruzzese
giovannia@ru.is

Threat Model – Knowledge

The knowledge is identical to that of the original threat model (i.e., $\mathcal{K}$), but it must be assumed that the attacker expects that the ML-NIDS would detect the strategy $\mathcal{S}$ the attacker would follow if they did not apply any HsP

Otherwise, why would the attacker even attempt using an HsP?

Giovanni Apruzzese
giovannia@ru.is

Threat Model – Capabilities

The capabilities must not change w.r.t. $\mathcal{C}$: the attacker cannot be assumed to, e.g., control more hosts, or have more privileges on their already controlled hosts.

Otherwise, there would be a logical fallacy in the threat model

Giovanni Apruzzese
giovannia@ru.is

Threat Model – Strategy

The strategy *must* change w.r.t. $\mathcal{S}$, and is still dictated by the assumed $\mathcal{K}$ and $\mathcal{C}$, and must only include operations that involve the attacker's controlled hosts.

E.g., an attacker cannot "bribe" an employee to obtain a password; or physically go to a host they wish to portscan in order to check which ports are open locally.

Giovanni Apruzzese
giovannia@ru.is

Research Question 1

“Has prior work on the robustness of ML-NIDS considered perturbations involving launching different commandson the attacker-controlled host (i.e., HsP)?”

Giovanni Apruzzese
giovannia@ru.is

Research Question 1

"Has prior work on the robustness of ML-NIDS considered perturbations involving launching different commandson the attacker-controlled host (i.e., HsP)?"

- Method: systematic literature review (n=316)
- Findings: no paper allows us to answer "yes" to RQ1
- Notable mentions: [32] and [18]

[32]: Catillo, Marta, et al. "Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection." Proceedings of the 19th International Conference on Availability, Reliability and Security. 2024.
[18]: Apruzzese, Giovanni, Aurore Fass, and Fabio Pierazzi. "When adversarial perturbations meet concept drift: an exploratory analysis on ml-nids." Proceedings of the 2024 Workshop on Artificial Intelligence and Security. 2024.

Giovanni Apruzzese
giovannia@ru.is

Research Question 1

"Has prior work on the robustness of ML-NIDS considered perturbations involving launching different commandson the attacker-controlled host (i.e., HsP)?"

- Method: systematic literature review (n=316)
- Findings: no paper allows us to answer "yes" to RQ1
- Notable mentions: [32] and [18]

Studying HsP requires performing actions on physical hosts

[32]: Catillo, Marta, et al. "Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection." Proceedings of the 19th International Conference on Availability, Reliability and Security. 2024.
[18]: Apruzzese, Giovanni, Aurore Fass, and Fabio Pierazzi. "When adversarial perturbations meet concept drift: an exploratory analysis on ml-nids." Proceedings of the 2024 Workshop on Artificial Intelligence and Security. 2024.

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

- Method: experiments on benchmark datasets, and on a real-world network
- Attacker's Goal: SSH-bruteforcing, particularly by using *ssh-patator* [5]
- ML-NIDS: trained on *ssh-patator --persistent=1* (or *--persistent=0*)
- Attacker's Strategy (HsP): *ssh-patator --persistent=0* (or *--persistent=1*)

[5]: https://github.com/lanjelot/patator

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

- Method: experiments on benchmark datasets, and on a real-world network
- Attacker's Goal: SSH-bruteforcing, particularly by using *ssh-patator* [5]
- ML-NIDS: trained on *ssh-patator --persistent=1* (or *--persistent=0*)
- Attacker's Strategy (HsP): *ssh-patator --persistent=0* (or *--persistent=1*)

Dataset	CICIDS17		CICIDS18
Train Set	Benign + P=1	Benign + all malicious	Benign + P=1
Test Set			
DT			
RF			
XGB			
SVM			
DNN			

Benchmarks

Train Set	Benign + P=1	Benign + P=0
Test Set		
DT		
RF		
XGB		
SVM		
DNN		

Real World

[5]: https://github.com/lanjelot/patator

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

- Method: experiments on benchmark datasets, and on a real-world network
- Attacker's Goal: SSH-bruteforcing, particularly by using *ssh-patator* [5]
- ML-NIDS: trained on *ssh-patator --persistent=1* (or *--persistent=0*)
- Attacker's Strategy (HsP): *ssh-patator --persistent=0* (or *--persistent=1*)

Dataset	CICIDS17				CICIDS18	
Train Set	Benign + P=1		Benign + all malicious		Benign + P=1	
Test Set	Benign	P=1	Benign	(all)	Benign	P=1
DT	<0.001	0.997	<0.001	>0.999	0.000	1.000
RF	0.000	0.998	<0.001	>0.999	0.000	1.000
XGB	<0.001	0.998	<0.001	>0.999	0.000	1.000
SVM	<0.001	0.993	<0.001	0.997	0.000	1.000
DNN	0.000	0.998	<0.001	>0.999	0.000	1.000

Benchmarks

Train Set	Benign + P=1		Benign + P=0	
Test Set	Benign	P=1	Benign	P=0
DT	<0.001	1.000	<0.001	1.000
RF	0.000	1.000	0.000	1.000
XGB	0.000	>0.999	<0.001	1.000
SVM	0.000	1.000	0.000	1.000
DNN	<0.001	1.000	<0.001	1.000

Real World

[5]: https://github.com/lanjelot/patator

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

- Method: experiments on benchmark datasets, and on a real-world network
- Attacker's Goal: SSH-bruteforcing, particularly by using *ssh-patator* [5]
- ML-NIDS: trained on *ssh-patator --persistent=1* (or *--persistent=0*)
- Attacker's Strategy (HsP): *ssh-patator --persistent=0* (or *--persistent=1*)

Dataset	CICIDS17						CICIDS18		
Train Set	Benign + P=1			Benign + all malicious			Benign + P=1		
Test Set	Benign	P=1	**P=0**	Benign	(all)	**P=0**	Benign	P=1	**P=0**
DT	<0.001	0.997	0.224	<0.001	>0.999	0.214	0.000	1.000	0.000
RF	0.000	0.998	0.490	<0.001	>0.999	0.073	0.000	1.000	0.000
XGB	<0.001	0.998	0.986	<0.001	>0.999	<0.001	0.000	1.000	0.000
SVM	<0.001	0.993	0.000	<0.001	0.997	<0.001	0.000	1.000	0.000
DNN	0.000	0.998	0.000	<0.001	>0.999	<0.001	0.000	1.000	0.000

Benchmarks

Train Set	Benign + P=1			Benign + P=0		
Test Set	Benign	P=1		Benign	P=0	
DT	<0.001	1.000		<0.001	1.000	
RF	0.000	1.000		0.000	1.000	
XGB	0.000	>0.999		<0.001	1.000	
SVM	0.000	1.000		0.000	1.000	
DNN	<0.001	1.000		<0.001	1.000	

Real World

[5]: https://github.com/lanjelot/patator

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

- Method: experiments on benchmark datasets, and on a real-world network
- Attacker's Goal: SSH-bruteforcing, particularly by using *ssh-patator* [5]
- ML-NIDS: trained on *ssh-patator --persistent=1* (or *--persistent=0*)
- Attacker's Strategy (HsP): *ssh-patator --persistent=0* (or *--persistent=1*)

Dataset	CICIDS17						CICIDS18		
Train Set	Benign + P=1			Benign + all malicious			Benign + P=1		
Test Set	Benign	P=1	**P=0**	Benign	(all)	**P=0**	Benign	P=1	**P=0**
DT	<0.001	0.997	0.224	<0.001	>0.999	0.214	0.000	1.000	0.000
RF	0.000	0.998	0.490	<0.001	>0.999	0.073	0.000	1.000	0.000
XGB	<0.001	0.998	0.986	<0.001	>0.999	<0.001	0.000	1.000	0.000
SVM	<0.001	0.993	0.000	<0.001	0.997	<0.001	0.000	1.000	0.000
DNN	0.000	0.998	0.000	<0.001	>0.999	<0.001	0.000	1.000	0.000

Benchmarks

Train Set	Benign + P=1			Benign + P=0		
Test Set	Benign	P=1	P=0	Benign	P=0	P=1
DT	<0.001	1.000	0.000	<0.001	1.000	0.000
RF	0.000	1.000	0.000	0.000	1.000	0.000
XGB	0.000	>0.999	0.000	<0.001	1.000	0.000
SVM	0.000	1.000	0.490	0.000	1.000	1.000
DNN	<0.001	1.000	0.000	<0.001	1.000	1.000

Real World

[5]: https://github.com/lanjelot/patator

Giovanni Apruzzese
giovannia@ru.is

Research Question 2

"What are some effects that HsP can have on existing ML-NIDS?"

ANSWER TO RQ2: Some HsP can drop the *tpr* of an ML-NIDS from 1 to 0 by changing one character in the command launched by the attacker's controlled host. Even HsP leveraging a different tool but for the same malicious goal can lead to complete bypass. HsP requiring higher privileges are not necessarily more evasive. The effectiveness of HsP against specific ML models varies.

Benchmarks

[5]: https://github.com/lanjelot/patator

Real World

Giovanni Apruzzese
giovannia@ru.is

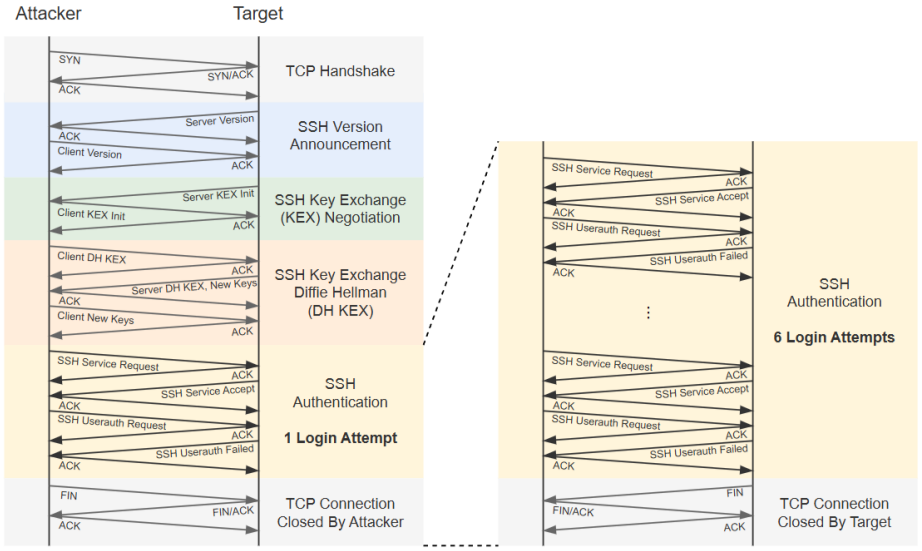
Explanation

“Why is it that, for *ssh-patator*, changing a single character can lead to a complete bypass of the ML-NIDS?”

Giovanni Apruzzese
giovannia@ru.is

Explanation

"Why is it that, for *ssh-patator*, changing a single character can lead to a complete bypass of the ML-NIDS?"

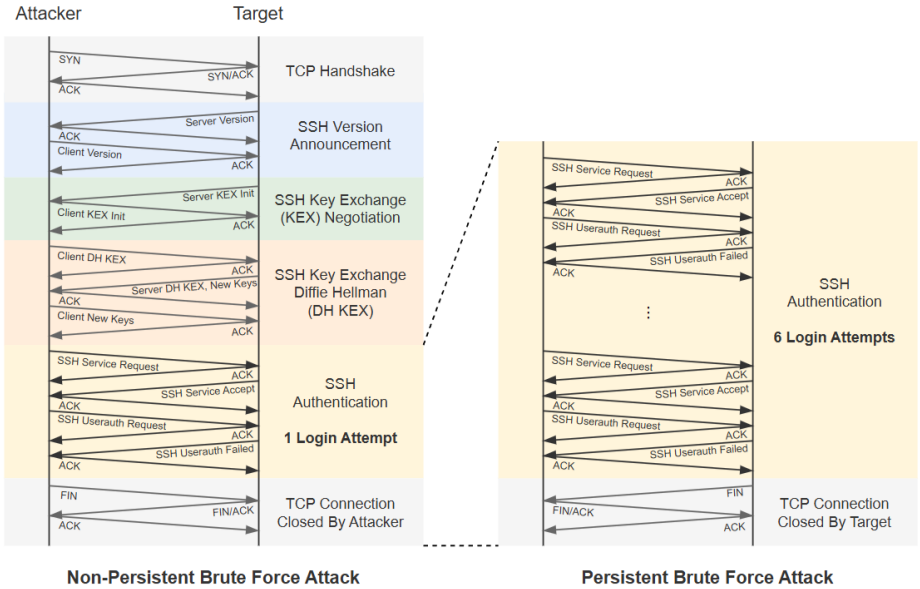


(a) Packet-level. By launching patator -P=0 (left), only one login attempt is made. However, launching patator -P=1 (right) leads to 6x more attempts. This difference is the "packet-level perturbation" induced by an HsP that changes -P=0 to patator -P=1 (i.e., a one-character change in the command launched by the attacker).

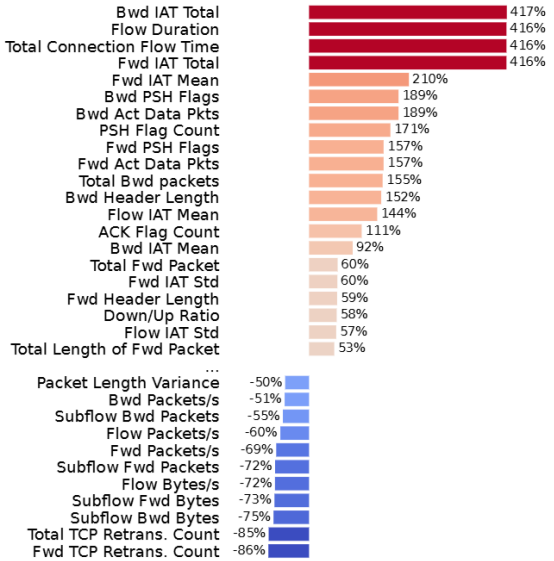
Giovanni Apruzzese
giovannia@ru.is

Explanation

"Why is it that, for *ssh-patator*, changing a single character can lead to a complete bypass of the ML-NIDS?"



(a) Packet-level. By launching patator -P=0 (left), only one login attempt is made. However, launching patator -P=1 (right) leads to 6x more attempts. This difference is the "packet-level perturbation" induced by an HsP that changes -P=0 to patator -P=1 (i.e., a one-character change in the command launched by the attacker).



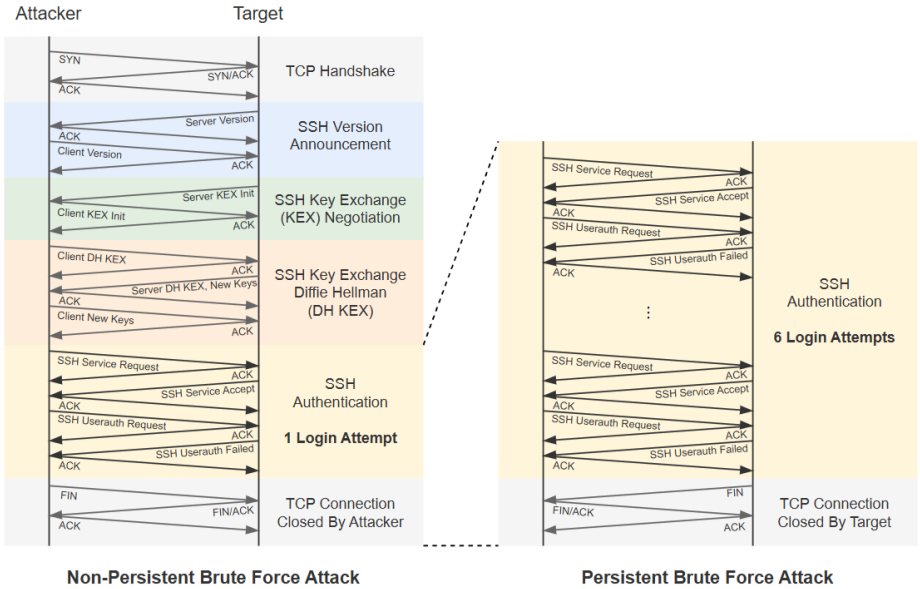
(b) NetFlow-level. In a the NetFlow-based feature space, an HsP switching -P=0 to -P=1 leads to substantial differences (numbers in the plot are obtained by computing -P=0 / -P=1)

Giovanni Apruzzese
giovannia@ru.is

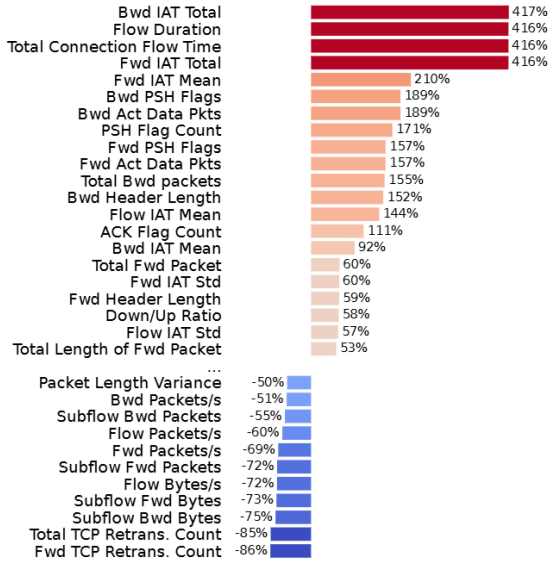
Exp

TAKEAWAY: Changing a single option (or parameter value) of a malicious command can lead to substantially different network activities—which achieve the same goal, but can cause misclassifications due to OOD (exploitable via HsP). Attempts to reproduce, or approximate, such changes by manipulating pre-collected datapoints is challenging and may result in misleading evaluations.

"Why … aracter …s?"



(a) Packet-level. By launching patator -P=0 (left), only one login attempt is made. However, launching patator -P=1 (right) leads to 6x more attempts. This difference is the "packet-level perturbation" induced by an HsP that changes -P=0 to patator -P=1 (i.e., a one-character change in the command launched by the attacker).



(b) NetFlow-level. In a the NetFlow-based feature space, an HsP switching -P=0 to -P=1 leads to substantial differences (numbers in the plot are obtained by computing -P=0 / -P=1)

Giovanni Apruzzese
giovannia@ru.is

Out of Distribution?

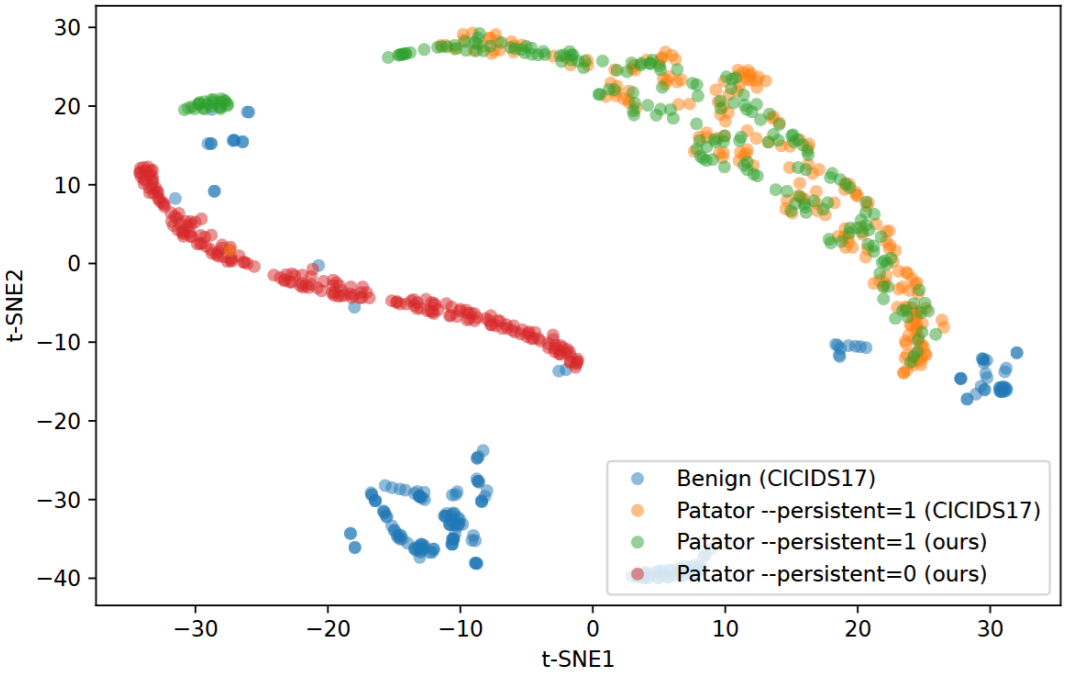


Fig. 4: OOD verification. We use a tSNE plot to visualize the distribution of NetFlows generated by patator (as well as benign ones).

Giovanni Apruzzese
giovannia@ru.is

But!

- Only one attack! → We show more attacks in the paper

Giovanni Apruzzese
giovannia@ru.is

But!

- Only one attack! → We show more attacks in the paper
- Only one model type! → We evaluate other model types in the paper!

Giovanni Apruzzese
giovannia@ru.is

But!

- Only one attack! → We show more attacks in the paper
- Only one model type! → We evaluate other model types in the paper!
- Only one benchmark dataset! → We evaluate also in other datasets

Giovanni Apruzzese
giovannia@ru.is

But!

- Only one attack! → We show more attacks in the paper
- Only one model type! → We evaluate other model types in the paper!
- Only one benchmark dataset! → We evaluate also in other datasets
- HsP are not adversarial perturbations! → Let’s discuss ☺

Giovanni Apruzzese
giovannia@ru.is

Conclusions

- HsP describe “alternative strategies” to bypass (ML-)NIDS
- Prior work on robustness of ML-NIDS studied “hard-to-realise” adversarial perturbations, but we showed that evading ML-NIDS via HsP is simple
- Training on a single “form” of a given attack is risky

Giovanni Apruzzese
giovannia@ru.is

Conclusions – so what now?

- HsP describe “alternative strategies” to bypass (ML-)NIDS
- Prior work on robustness of ML-NIDS studied “hard-to-realise” adversarial perturbations, but we showed that evading ML-NIDS via HsP is simple
- Training on a single “form” of a given attack is risky
- Future work should investigate “easy-to-apply” perturbations implemented at the host level (i.e., HsP)
- (hard problem) mapping fine-grained feature-space perturbations to HsP

Giovanni Apruzzese
giovannia@ru.is

Bangalore, June 4th 2026

21st ACM ASIA Conference on Computer and Communications Security

“What is the Problem Space?” Defining Host-space Adversarial Perturbations against Network Intrusion Detection Systems

Miel Verkerken, Laurens d’Hooge, Bruno Volckaert, Filip de Turck, Giovanni Apruzzese

HÁSKÓLINN Í REYKJAVÍK | REYKJAVIK UNIVERSITY

IEEE SaTML'26

ConCap: Practical Network Traffic Generation for (ML- and) Flow-based Intrusion Detection Systems

Miel Verkerken[†], Laurens D'hooge[†], Bruno Volckaert[†], Filip De Turck[†], Giovanni Apruzzese[¶]
[†]*Ghent University – imec*, [¶]*University of Liechtenstein*, [¶]*Reykjavik University*,
{name.surname}@{[†]ugent.be, [¶]uni.li}