



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

What if the attackers are inside?

A Systematic Analysis of Label-flipping Attacks against Federated Learning for Intrusion Detection

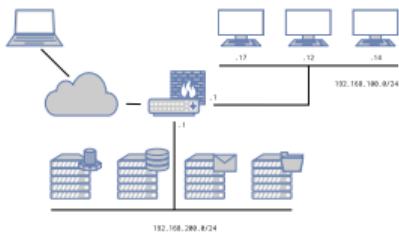
Léo Lavaur¹, Yann Busnel², and Fabien Autrel¹

¹ IMT Atlantique, ² IMT Nord Europe

ARES (BASS) 2024, August 2, 2024

Network-based Intrusion Detection System (NIDS)

1. Data collection



2. Data processing

pkt_pkts	pkt_sum	pkt_len	pkt_rtt
0.4545	0.69856	0.36146	0.32256
0.1641	0.387	0.39	0.32184
0.1684	0.0126	0.9774	0.3754
0.454	0.42164	0.39974	0.187
0.36845	0.4175	0.14034	0.186
0.00214	0.265844876	0.332	0.48
0.265	0.8565453	0.1548764	0.554935
0.15406	0.3021181	0.35493	0.454
0.36994	0.4987	0.3484	0.34094
0.2021521	0.20	0.151180	0.196495

3. Data labeling

pkt_pkts	pkt_sum	pkt_len	pkt_rtt	LABEL
0.4545	0.69856	0.36146	0.32256	Benign
0.1641	0.387	0.39	0.32184	Malicious
0.1684	0.0126	0.9774	0.3754	Benign
0.454	0.42164	0.39974	0.187	Malicious
0.36845	0.4175	0.14034	0.186	Malicious
0.00214	0.265844876	0.332	0.48	Benign
0.265	0.8565453	0.1548764	0.554935	Benign
0.15406	0.3021181	0.35493	0.454	Malicious
0.36994	0.4987	0.3484	0.34094	Malicious
0.2021521	0.20	0.151180	0.196495	Malicious

4. Model training

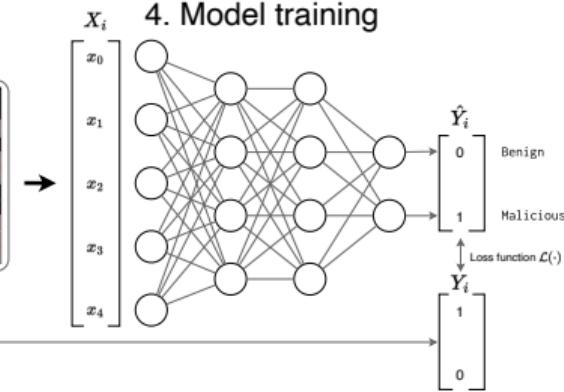
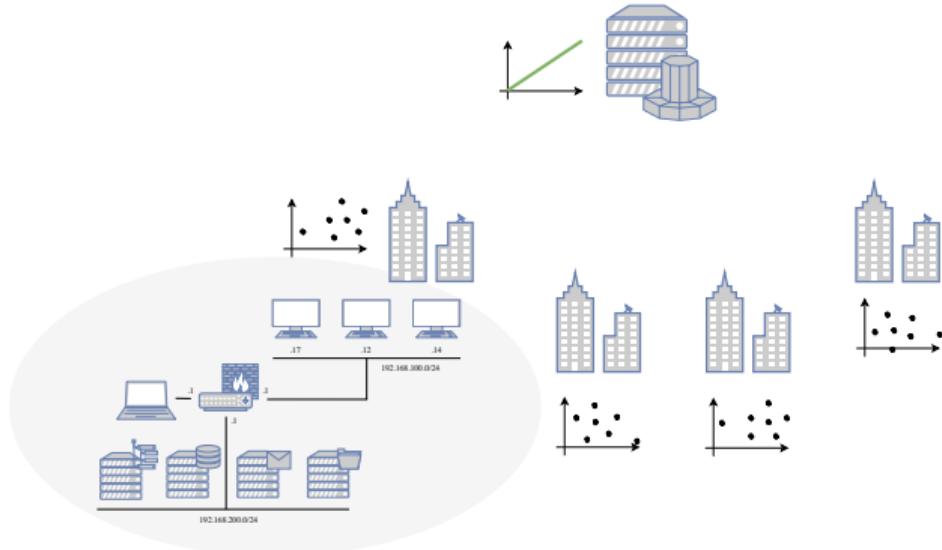


Figure: Typical NIDS workflow.

- ▶ Great performance with Deep Learning (DL) (on public datasets at least)
- ▶ **Limitations:** lack of labelled data, risk of local bias or skewed data distribution, inefficient against new attacks.

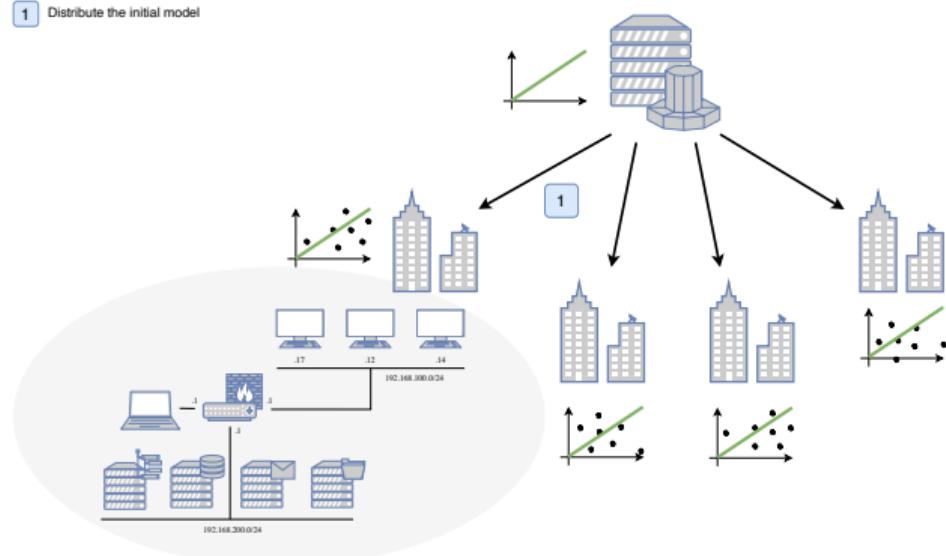
Scaling NIDS with FL

- ▶ Federated Learning (FL) is a distributed Machine Learning (ML) paradigm [1].
- ▶ Participants train a global model without sharing local data.



Scaling NIDS with FL

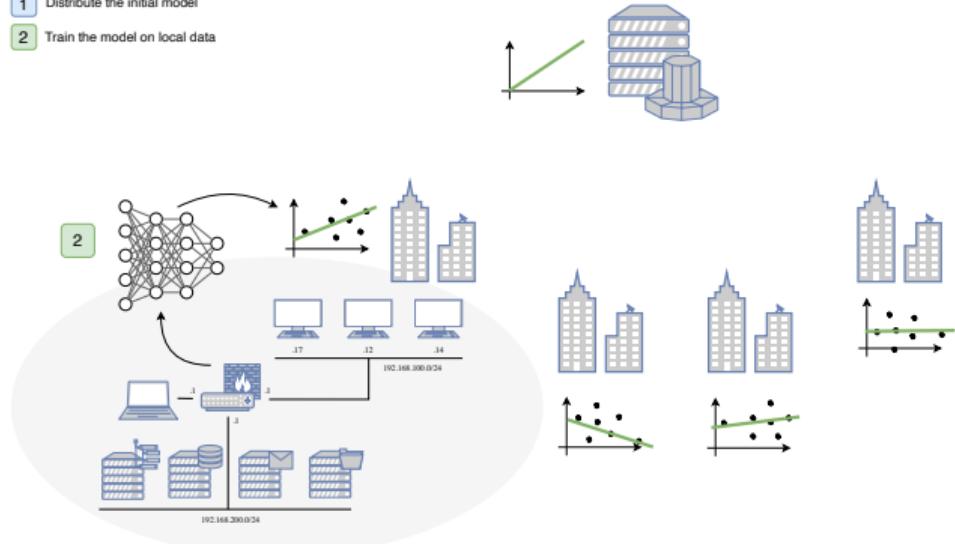
- ▶ Federated Learning (FL) is a distributed Machine Learning (ML) paradigm [1].
- ▶ Participants train a global model without sharing local data.



Scaling NIDS with FL

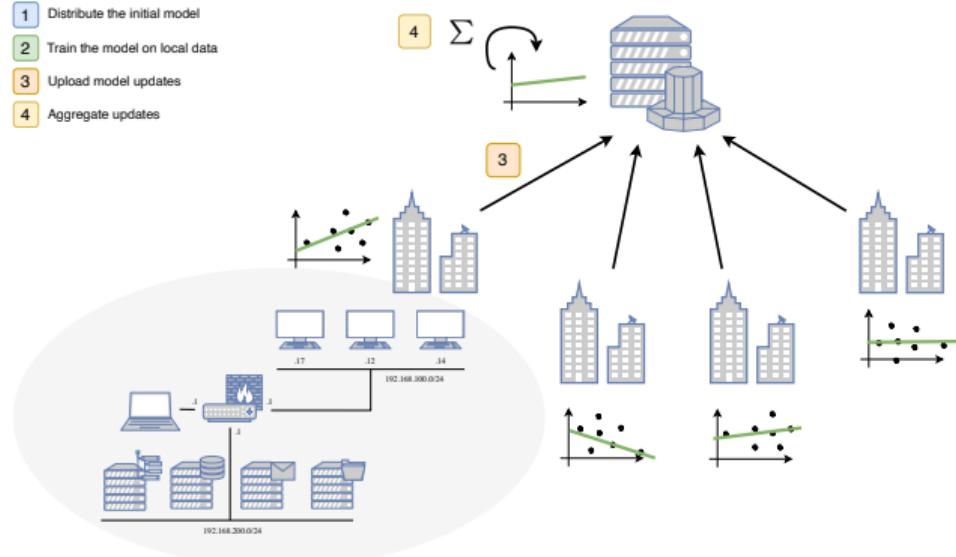
- ▶ Federated Learning (FL) is a distributed Machine Learning (ML) paradigm [1].
- ▶ Participants train a global model without sharing local data.

- 1 Distribute the initial model
- 2 Train the model on local data



Scaling NIDS with FL

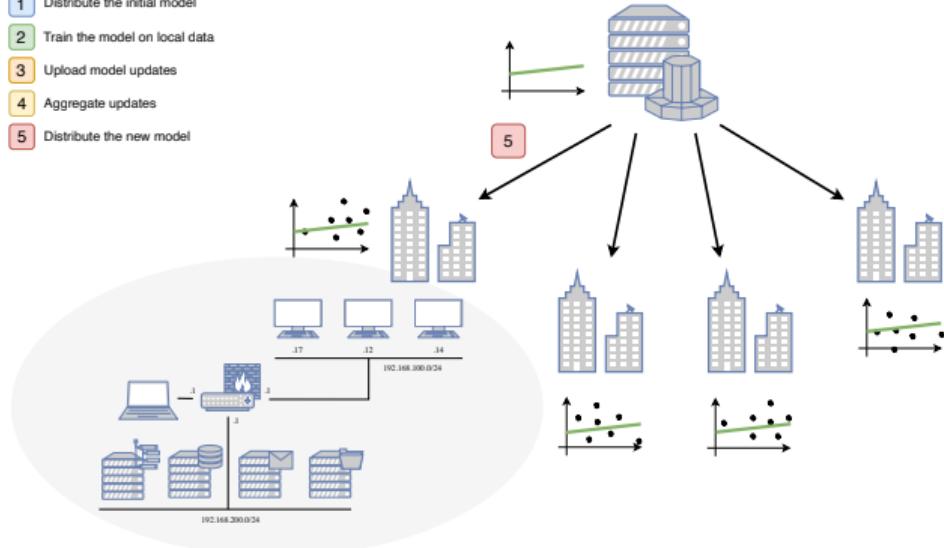
- ▶ Federated Learning (FL) is a distributed Machine Learning (ML) paradigm [1].
- ▶ Participants train a global model without sharing local data.



Scaling NIDS with FL

- ▶ Federated Learning (FL) is a distributed Machine Learning (ML) paradigm [1].
- ▶ Participants train a global model without sharing local data.

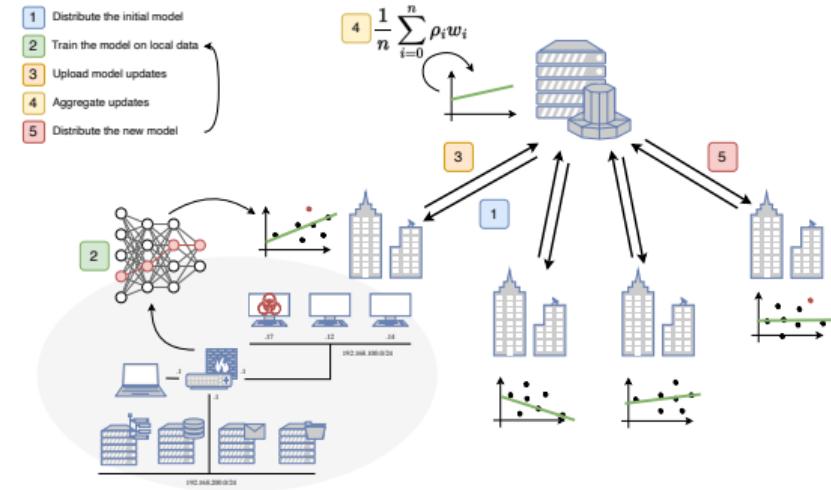
- 1 Distribute the initial model
- 2 Train the model on local data
- 3 Upload model updates
- 4 Aggregate updates
- 5 Distribute the new model



Federated Intrusion Detection System (FIDS)

FL can be used in Collaborative Intrusion Detection System (CIDS) [2]:

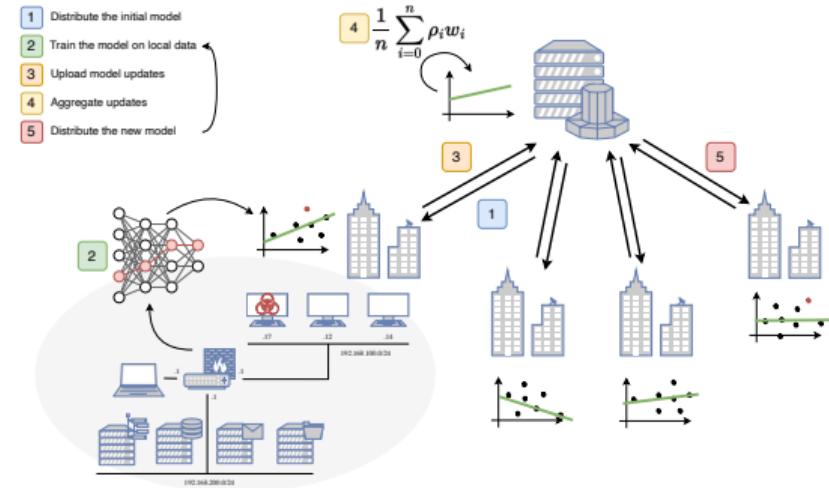
- ▶ Extend the training data with Horizontal Federated Learning (HFL)
 - Reduce the risk of local bias



Federated Intrusion Detection System (FIDS)

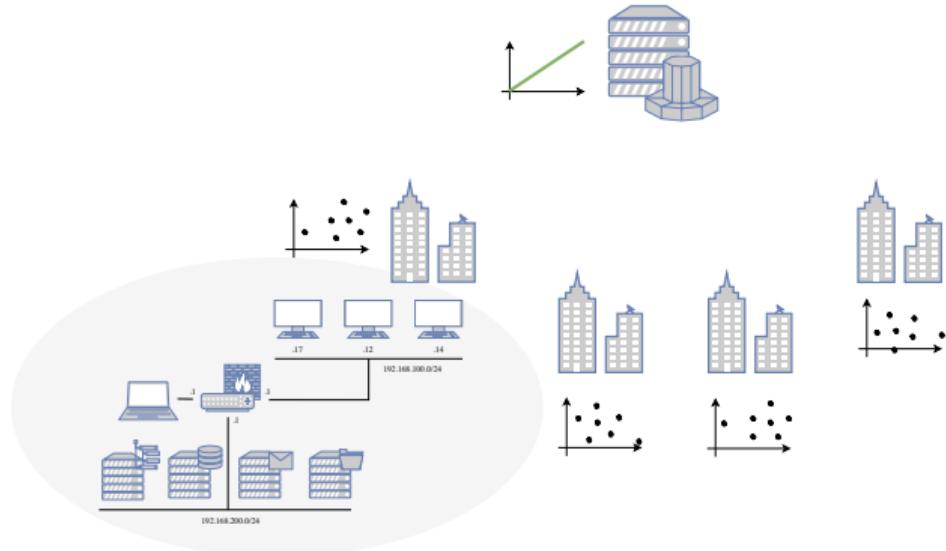
FL can be used in Collaborative Intrusion Detection System (CIDS) [2]:

- ▶ Extend the training data with Horizontal Federated Learning (HFL)
 - Reduce the risk of local bias
- ▶ Effectively share knowledge (e.g., on specific classes, instances) between participants
 - Share the knowledge about a new attack [3];
 - Improve the characterization of specific devices; ...



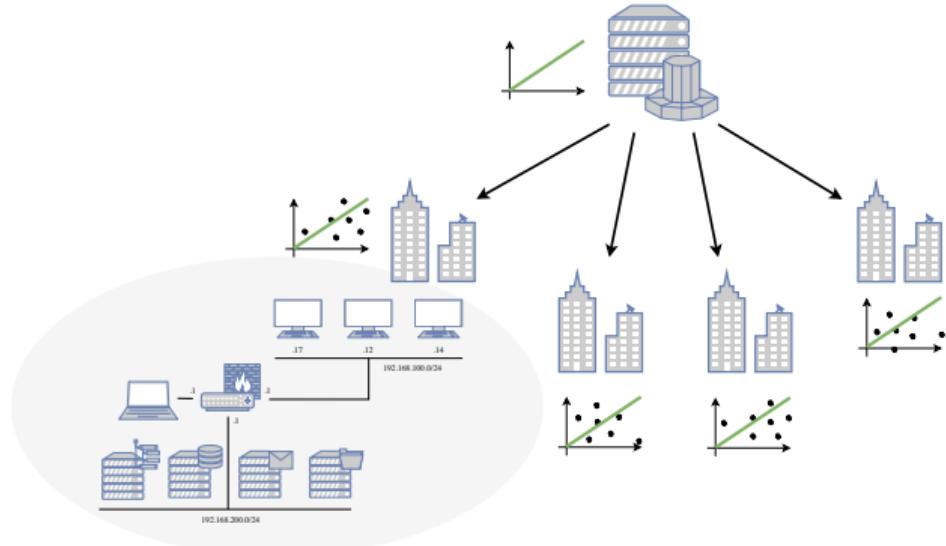
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].



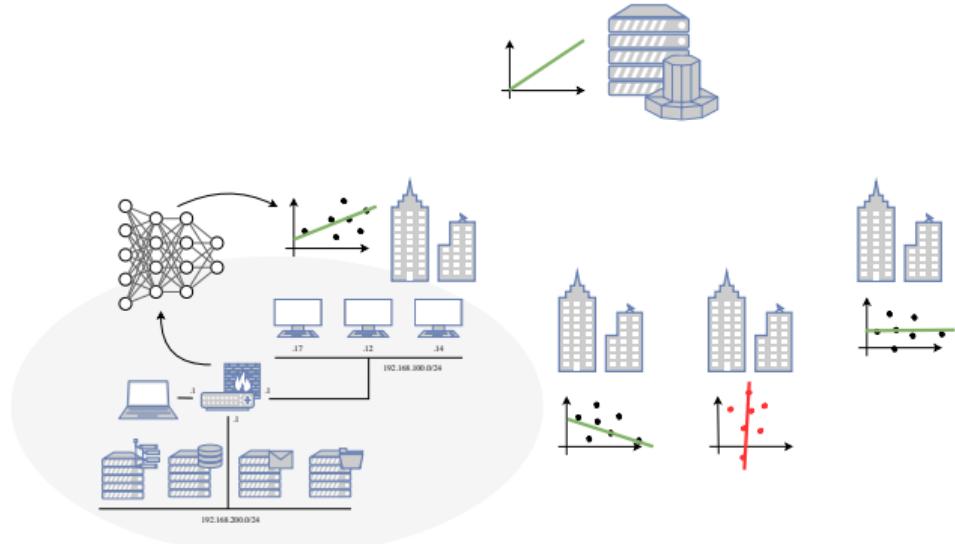
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].



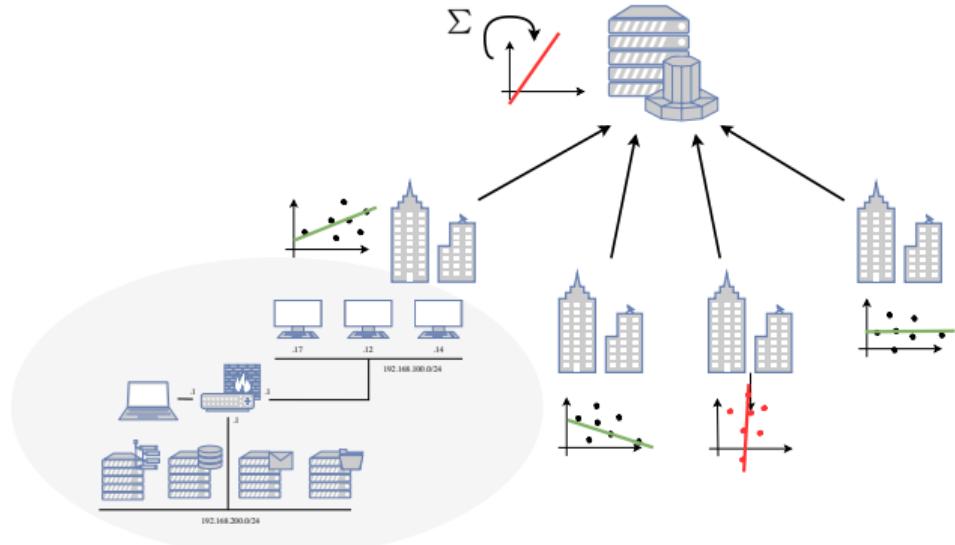
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].



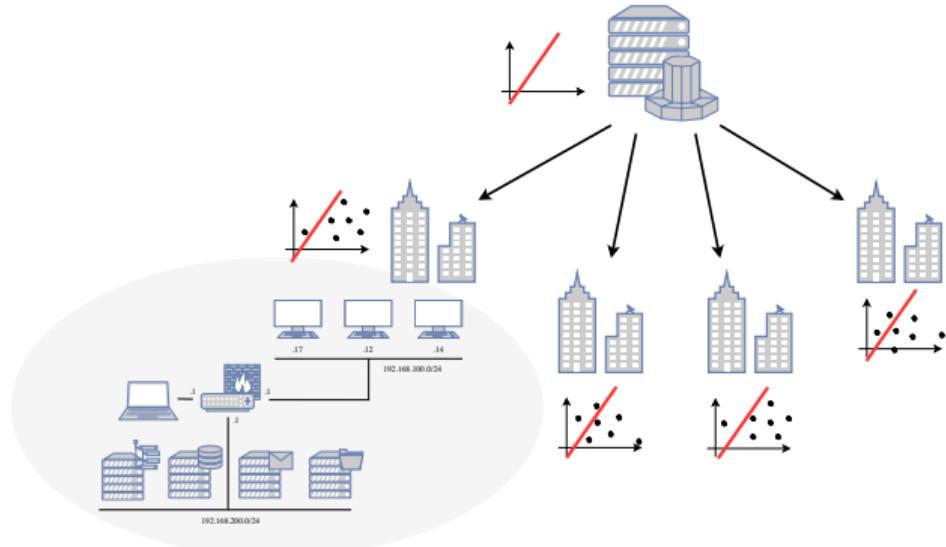
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].



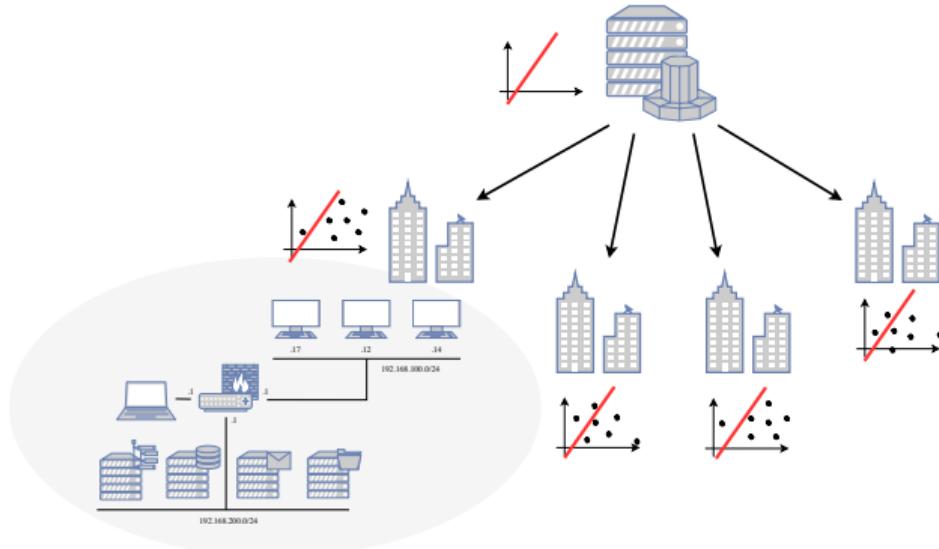
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].



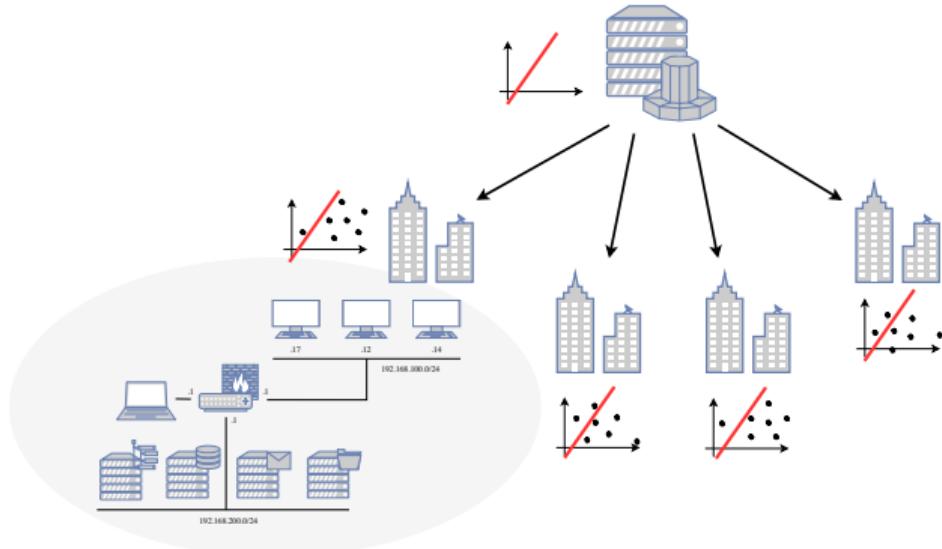
FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].
- ▶ Lots of existing countermeasures [5]–[7] in FIDS already...



FL against malicious contributions

- ▶ FL is highly susceptible to poisoning [4].
- ▶ Lots of existing countermeasures [5]–[7] in FIDS already...
- ▶ Yet, their impact is difficult to estimate.
 - Few studies in the FIDS context, often partial.
 - Nothing on aggregated classes.



Types of poisoning attacks

- ▶ By component:
 - Data poisoning (e.g., label-flipping, clean-label attacks, backdoors)
 - Model poisoning (e.g., gradient boosting, noising)
- ▶ By target:
 - Untargeted: affect the model's global performance
 - Targeted: modify its behavior on specific classes or instances
- ▶ By frequency:
 - one-shot: attacks are performed once
 - iterative/continuous: at each round
 - adaptive: reacts to the model aggregation

Types of poisoning attacks

- ▶ By component:
 - Data poisoning (e.g., **label-flipping**, clean-label attacks, backdoors)
 - Model poisoning (e.g., gradient boosting, noising)
- ▶ By target:
 - **Untargeted**: affect the model's global performance
 - **Targeted**: modify its behavior on specific classes or instances
- ▶ By frequency:
 - one-shot: attacks are performed once
 - **iterative/continuous: at each round**
 - adaptive: reacts to the model aggregation

Our work

Continuous label-flipping attacks in collaborative Intrusion Detection System (IDS) context.

OUTLINE

1. Methodology and Research Questions
2. Experimental setup
3. Results
4. Conclusion

OUTLINE

1. Methodology and Research Questions
2. Experimental setup
3. Results
4. Conclusion

Threat model

- ▶ A malicious participant can alter its local dataset before training, and start/stop the attack at any round r .
 - Either done by the client itself or by an external attacker.

Threat model

- ▶ A malicious participant can alter its local dataset before training, and start/stop the attack at any round r .
 - Either done by the client itself or by an external attacker.
- ▶ The attacker can manipulate labels of its dataset.
 - Untargeted: flip the labels of a proportion of samples.
 - Targeted: associate benign labels to a proportion of samples from a specific attack class.

Threat model

- ▶ A malicious participant can alter its local dataset before training, and start/stop the attack at any round r .
 - Either done by the client itself or by an external attacker.
- ▶ The attacker can manipulate labels of its dataset.
 - Untargeted: flip the labels of a proportion of samples.
 - Targeted: associate benign labels to a proportion of samples from a specific attack class.
- ▶ An attacker can poison any proportion of its local data
 - Data Poisoning Rate (DPR): $(\alpha) \rightarrow$ proportion of flipped samples.

Threat model

- ▶ A malicious participant can alter its local dataset before training, and start/stop the attack at any round r .
 - Either done by the client itself or by an external attacker.
- ▶ The attacker can manipulate labels of its dataset.
 - Untargeted: flip the labels of a proportion of samples.
 - Targeted: associate benign labels to a proportion of samples from a specific attack class.
- ▶ An attacker can poison any proportion of its local data
 - Data Poisoning Rate (DPR): $(\alpha) \rightarrow$ proportion of flipped samples.
- ▶ Multiple attackers can collude to perform the attack.
 - Model Poisoning Rate (MPR): $(\tau) \rightarrow$ proportion of attackers.

Research questions

► Research questions (RQ):

- RQ1. Is the behavior of poisoning attacks predictable?
- RQ2. Are there beneficial or harmful combinations of hyperparameter under poisoning attacks?
- RQ3. Can FL heal itself from poisoning attacks?
- RQ4. Are IDS backdoors realistic using label-flipping attacks?
- RQ5. Is there a critical threshold where label-flipping attacks begin to impact performance?

Experiment orchestration using Eiffel [3].

- ▶ Flower simulation framework [8] for FL.
- ▶ Hydra [9] for experiment generation and configuration.
- ▶ Custom-made poisoning engine with different attack strategies.
- ▶ Nix [10] and Poetry to fix system and Python dependencies, enabling reproducibility.

Evaluation framework

Experiment orchestration using Eiffel [3].

- ▶ Flower simulation framework [8] for FL.
- ▶ Hydra [9] for experiment generation and configuration.
- ▶ Custom-made poisoning engine with different attack strategies.
- ▶ Nix [10] and Poetry to fix system and Python dependencies, enabling reproducibility.

Table: Experimental parameters.

Parameter	Values	Description
<i>batch_size</i>	32, 128, 512	Batch size (β)
<i>epochs</i>	100_10x10, 100_4x25, 100_1x100, 300_10x30, 300_4x75, 300_1x300	Local epochs per round (\mathcal{E})
<i>distribution</i>	10-0, 9-1, 7-3, 5-5, 3-7	Proportion of attackers (τ)
<i>scenario</i>	continuous-[10,30,60,70,80,90,95,99], continuous-100, late-3, redemption-3	Poisoning rate per round (α)
<i>target</i>	untargeted, bot, dos, ddos, bruteforce, infiltration, injection	Attack type and target
<i>seed</i>	1313, 1977, 327, 5555, 501, 421, 3263827, 2187, 1138, 6567	Seed for PRNG

OUTLINE

1. Methodology and Research Questions
2. Experimental setup
3. Results
4. Conclusion

- ▶ Used dataset: sampled NF-V2 version of CSE-CIC-IDS2018
 - ports and IP addresses are removed
- ▶ Same class distribution in the training and testing sets
 - 80% of the dataset is used for training
 - 20% of the dataset is used for testing

- ▶ Used dataset: **sampled** NF-V2 version of CSE-CIC-IDS2018
 - ports and IP addresses are removed
- ▶ Same class distribution in the training and testing sets
 - 80% of the dataset is used for training
 - 20% of the dataset is used for testing
- ▶ We ensure the representativity of the dataset sampling

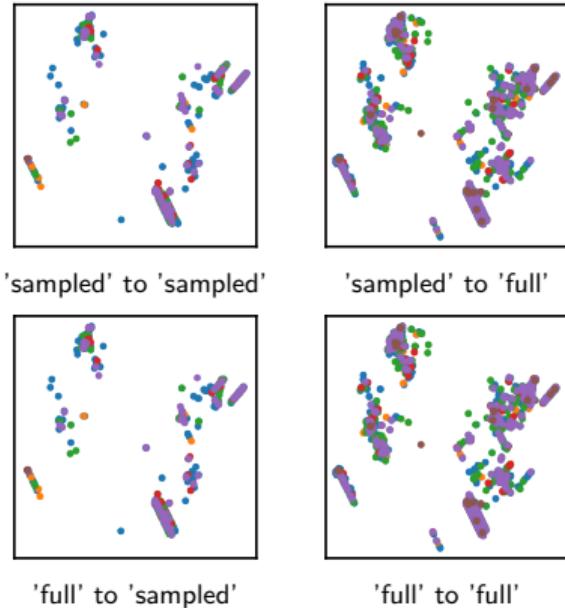


Figure: Cross-projection of the malicious traffic from two datasets in two dimensions using PCA.

- ▶ A simple multilayer perceptron (MLP) model with two hidden layers is used
- ▶ Baseline (centralized training)
 - F1-score: 0.966
 - Accuracy: 0.992
- ▶ FL setup
 - Cross-silo setting: all clients are available at each round
 - The dataset is partitioned into 10 Independent and Identically Distributed (IID) shards of 80,000 data points
 - Models are aggregated using FedAvg

Absolute Attack Success Rate (AASR)

- ▶ Targeted attacks → miss rate on the targeted class:

$$\frac{FN_{\text{class}}}{TP_{\text{class}} + FN_{\text{class}}}$$

- ▶ Untargeted attacks → misclassification rate:

$$1 - \text{accuracy}$$

Appropriate metrics

Absolute Attack Success Rate (AASR)

- ▶ Targeted attacks → miss rate on the targeted class:

$$\frac{FN_{\text{class}}}{TP_{\text{class}} + FN_{\text{class}}}$$

- ▶ Untargeted attacks → misclassification rate:

$$1 - \text{accuracy}$$

Relative Attack Success Rate (RASR)

- ▶ ASR relative to the benign scenario:

$$\frac{\max(AASR_{\text{benign}}, AASR_{\text{attack}}) - AASR_{\text{benign}}}{1 - AASR_{\text{benign}}}$$

OUTLINE

1. Methodology and Research Questions
2. Experimental setup
3. Results
4. Conclusion

RQ1: Is the behavior of poisoning attacks predictable?

17

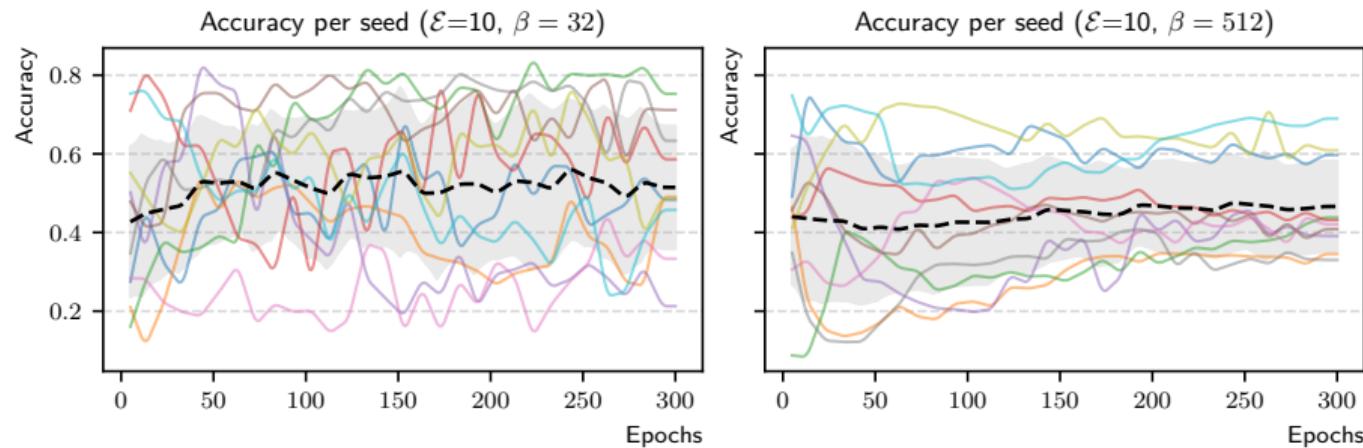


Figure: Accuracy of the poisoned model by seed.

RQ1: Is the behavior of poisoning attacks predictable?

18

Short answer:

RQ1: Is the behavior of poisoning attacks predictable?

18

Short answer: Nope.

RQ1: Is the behavior of poisoning attacks predictable?

18

Short answer: Nope.

Long answer:

- ▶ Very high variance in the results
- ▶ The impact of the attack is highly dependent on the seed
 - Initial parameters, data shuffling, ...
- ▶ Results tend to stabilize after a few rounds on different values

RQ2: Do hyperparameters influence the effect of poisoning attacks? 19

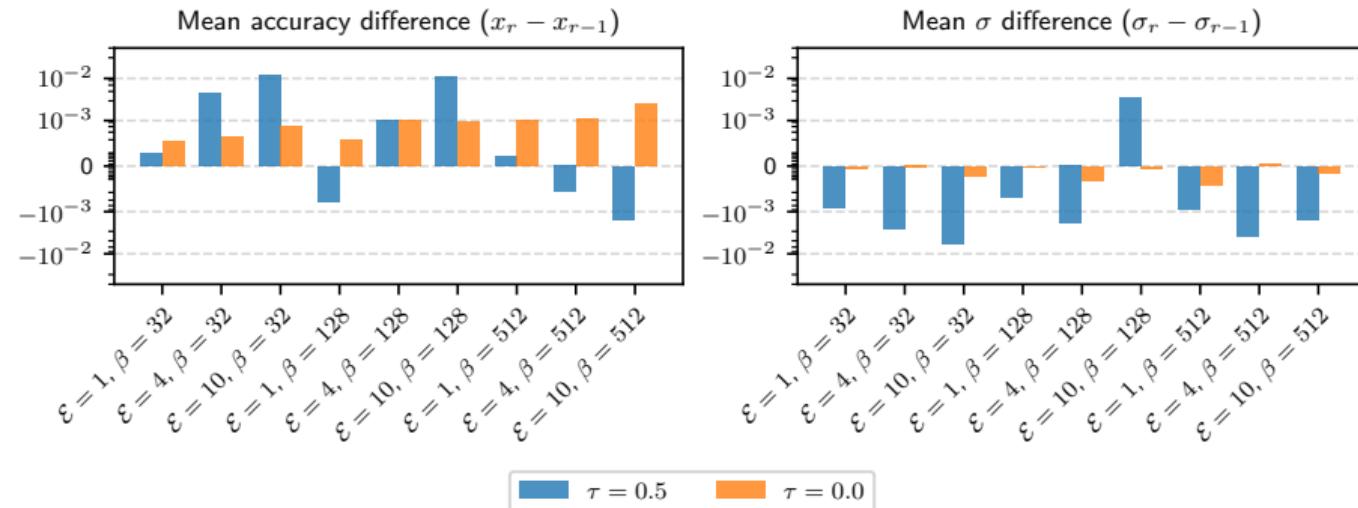


Figure: Effect of the hyperparameters on the accuracy of the poisoned model.

RQ2: Do hyperparameters influence the effect of poisoning attacks? 20

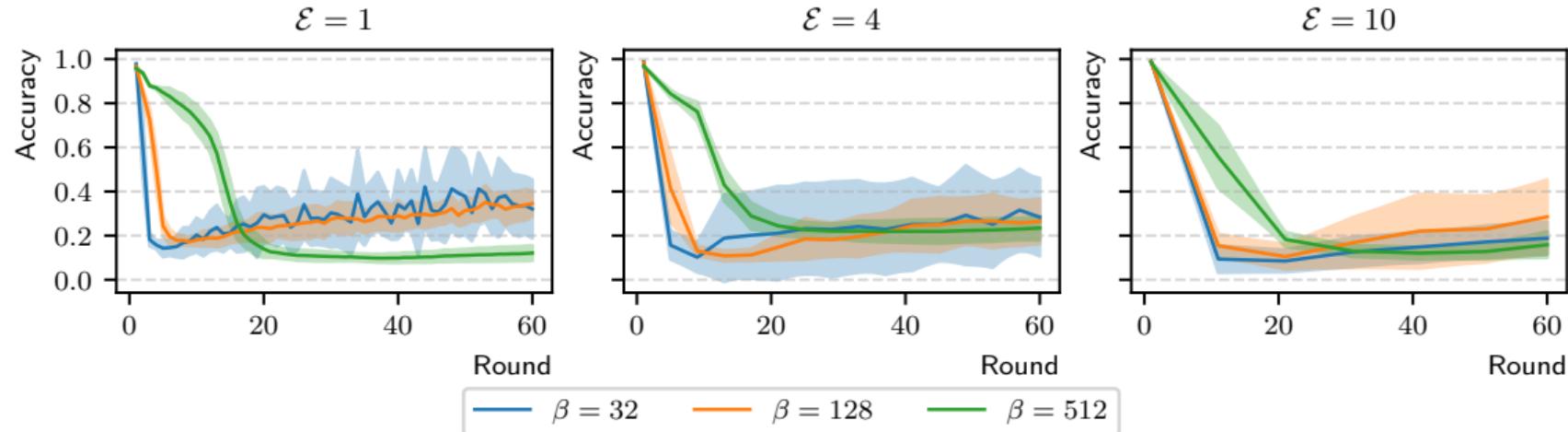


Figure: Effect of the hyperparameters on the accuracy of the poisoned model in the late scenario.

RQ2: Do hyperparameters influence the effect of poisoning attacks? 21

Answer:

- ▶ No impact on the average performance
- ▶ Significant impact on the variance of the results
- ▶ High batch size leads to more inertia
 - The impact is less instantaneous

RQ4: Are IDS backdoors realistic using label-flipping attacks?

22

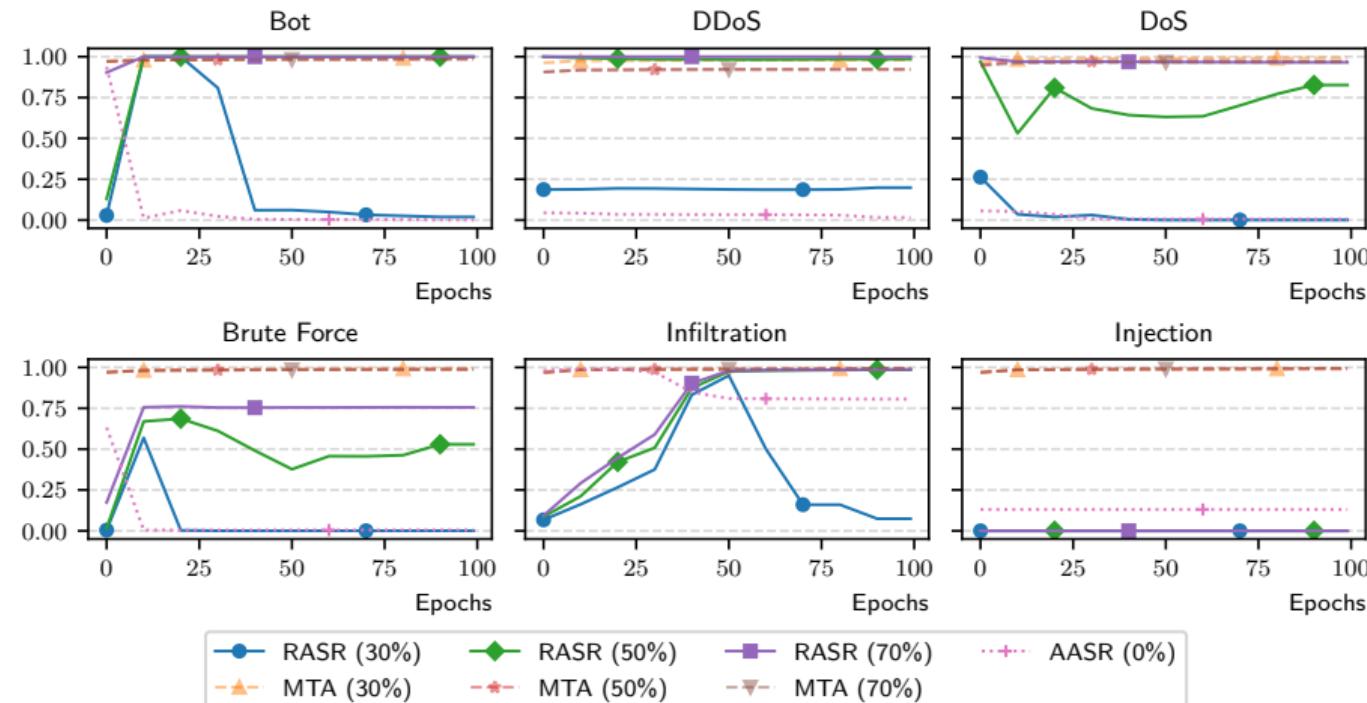


Figure: Backdoor success rate.

RQ4: Are IDS backdoors realistic using label-flipping attacks?

23

Answer:

- ▶ Yes, but...

Answer:

- ▶ Yes, but...
- ▶ The model's generalization capabilities can mitigate the impact
 - especially with characteristic overlaps between classes
- ▶ The attack's effectiveness is highly dependent on the target
- ▶ We need a significant number of attackers.

OUTLINE

1. Methodology and Research Questions
2. Experimental setup
3. Results
4. Conclusion

We build a reproducible framework to study the impact of label-flipping attacks in FIDS using FL, here are our main findings (yet):

- ▶ Label-flipping can be effective with enough attackers.
- ▶ Targeted label-flipping strives on well-detected targets.
 - but can be significantly mitigated by the model's generalization capabilities and the use of aggregated classes.
- ▶ Mitigation strategies must be adapted to the use case specificities (e.g., constrained environments).

→ *more detailed results in the paper*

- ▶ Extend the study to other datasets.
- ▶ Study the impact of the data distribution on the ability to detect attacks using similarity metrics.
- ▶ Build more appropriate metrics for the evaluation of poisoning attacks in Non Independent and Identically Distributed (NIID) settings.
- ▶ Extend to other feature sets and poisoning attacks.

- ▶ Extend the study to other datasets. [DONE]
- ▶ Study the impact of the data distribution on the ability to detect attacks using similarity metrics. [DONE]
- ▶ Build more appropriate metrics for the evaluation of poisoning attacks in NIID settings.
- ▶ Extend to other feature sets and poisoning attacks.
 - Our evaluation framework is generic enough (and open source!) to make extending the results easy.

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, Apr. 20–22, 2017, pp. 1273–1282. [Online]. Available: <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- [2] L. Lavaur, M.-O. Pahl, Y. Busnel, and F. Autrel, "The Evolution of Federated Learning-based Intrusion Detection and Mitigation: A Survey," *IEEE Transactions on Network and Service Management, Special Issue on Network Security Management*, Jun. 2022.
- [3] L. Lavaur, Y. Busnel, and F. Autrel, "Demo: Highlighting the limits of federated learning in intrusion detection," in *Proceedings of the 44th International Conference on Distributed Computing Systems (ICDCS)*, Jersey City, NJ, USA, Jul. 2024.
- [4] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data Poisoning Attacks Against Federated Learning Systems," in *Computer Security – ESORICS 2020*, L. Chen, N. Li, K. Liang, and S. Schneider, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 480–501, ISBN: 978-3-030-58951-6. DOI: 10.1007/978-3-030-58951-6_24.

- [5] N. C. Vy, N. H. Quyen, P. T. Duy, and V.-H. Pham, "Federated Learning-Based Intrusion Detection in the Context of IIoT Networks: Poisoning Attack and Defense," in *Network and System Security*, M. Yang, C. Chen, and Y. Liu, Eds., vol. 13041, Cham: Springer International Publishing, 2021, pp. 131–147, ISBN: 978-3-030-92707-3 978-3-030-92708-0. DOI: 10.1007/978-3-030-92708-0_8. [Online]. Available: https://link.springer.com/10.1007/978-3-030-92708-0_8 (visited on 03/05/2024).
- [6] R. Yang, H. He, Y. Wang, Y. Qu, and W. Zhang, "Dependable federated learning for IoT intrusion detection against poisoning attacks," *Computers & Security*, vol. 132, p. 103381, Sep. 1, 2023, ISSN: 0167-4048. DOI: 10.1016/j.cose.2023.103381. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404823002912> (visited on 03/04/2024).
- [7] Y. Zhang, Y. Zhang, Z. Zhang, H. Bai, T. Zhong, and M. Song, "Evaluation of data poisoning attacks on federated learning-based network intrusion detection system," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, Dec. 2022, pp. 2235–2242. DOI: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00330. [Online]. Available: <https://ieeexplore.ieee.org/document/10074658> (visited on 10/31/2023).

- [8] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020. arXiv: 2007.14390.
- [9] O. Yadan, *Hydra - A framework for elegantly configuring complex applications*, Github, 2019. [Online]. Available: <https://github.com/facebookresearch/hydra>.
- [10] E. Dolstra, "The purely functional software deployment model," s.n., S.I., 2006.

Thank you!

- ▶ Consider the use case when designing or choosing mitigations.
- ▶ Expect the unexpected: the behavior of poisoning attacks is unpredictable.
- ▶ Try it yourself! Our work is *reproducible* and everything is in *open-access*!

Paper



Results



Evaluation framework



Any questions?

RQ3: Can FL heal itself from poisoning attacks?

31

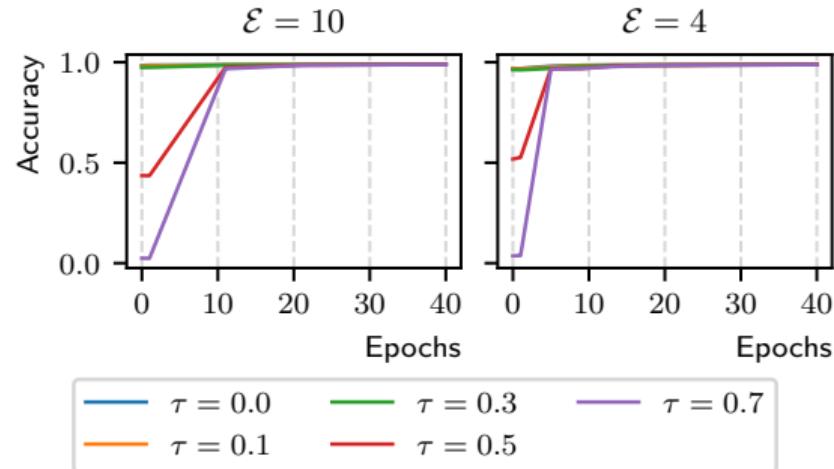


Figure: Recovery of the model after a poisoning attack.

RQ5: Is there a critical threshold for label-flipping to be effective?

32

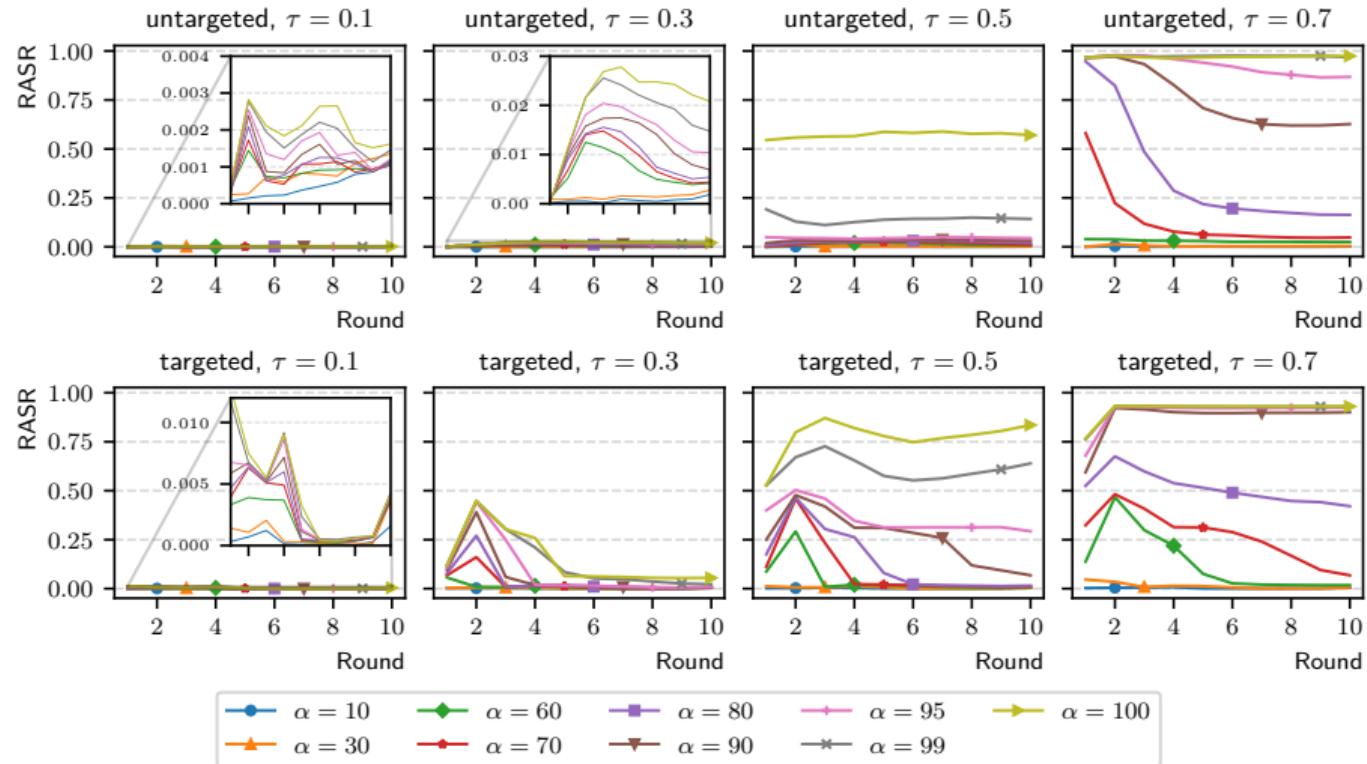


Figure: Impact of τ and α on the attack's effectiveness.