DiffuPac: Contextual Mimicry in Adversarial Packets Generation via Diffusion Model

ABDULLAH BIN JASNI

AKIKO MANADA

KOHEI WATABE

Background & Motivation

- **Importance of Network Intrusion Detection System (NIDS)**
  - Critical for detecting malicious activities in diverse applications (industrial, healthcare, etc.)
- **Emerging Threats**
  - Generative AI enables the creation of adversarial packets (concealing malicious intents), evading even advanced NIDS
- **Challenges in Existing Methods**
  - Traditional adversarial generation relies on unrealistic attacker knowledge (e.g., NIDS configuration)
- **Need for Innovation**
  - Practical solutions that consider the constraints faced by real attackers in ensuring robustness of NIDS in evolving threat landscapes

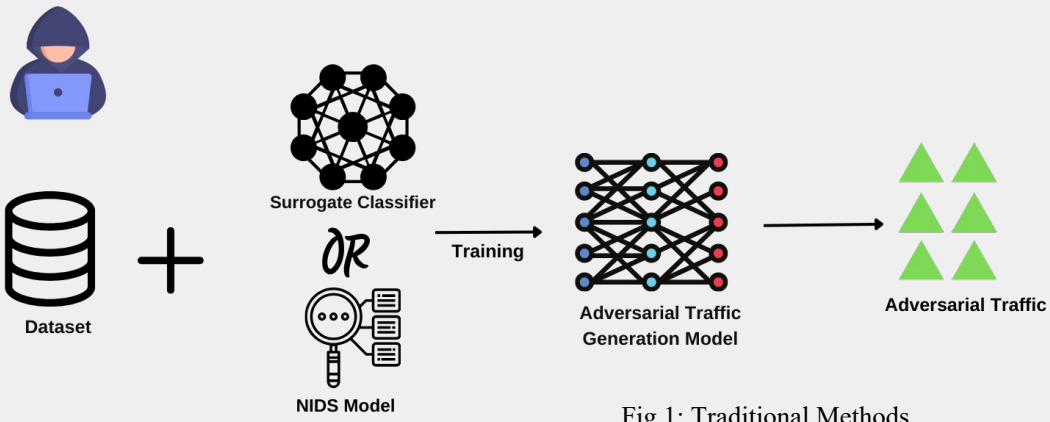


Fig.1: Traditional Methods

Principal Contribution

- DiffuPac model : Leverages the combined strengths of BERT and Diffusion model
  - Operates under the realistic assumption that attackers lack direct access to NIDS models
- Principal Contribution:
  - Pioneered the integration of BERT and diffusion models to create DiffuPac, marking a first in the cybersecurity domain
  - Introduce a unique concatenation strategy coupled with targeted noising technique
  - First adversarial packet generation framework to explicitly evaluate the malicious functionality, providing detailed insights into our evaluation process

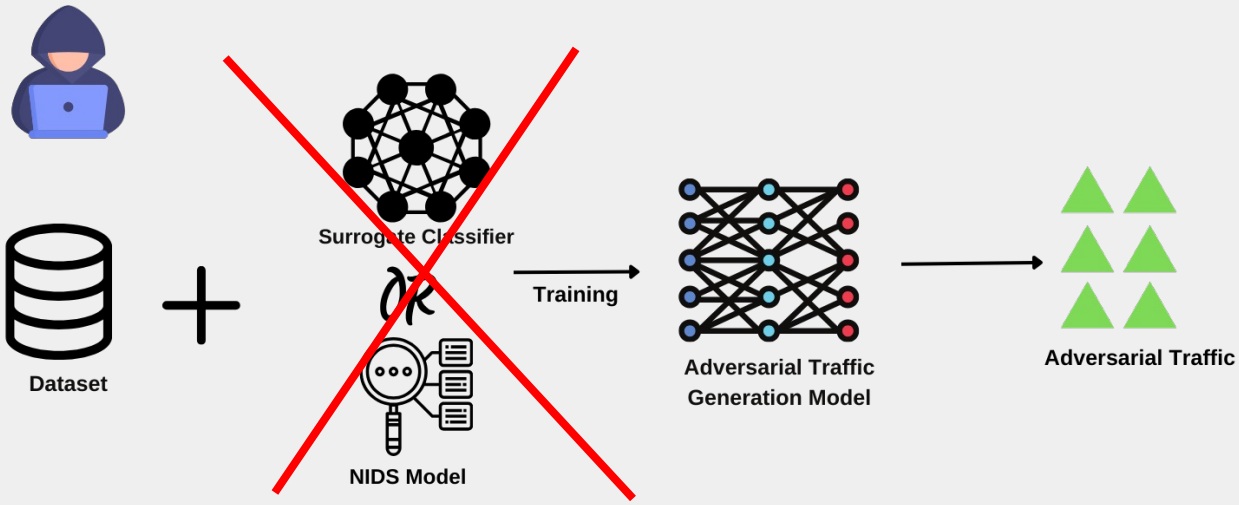


Fig. 2: New Method of Adversarial Generation

Proposed Architecture

- Divided into three phases: pcap pre-processing, pre-training and fine-tuning with diffusion models

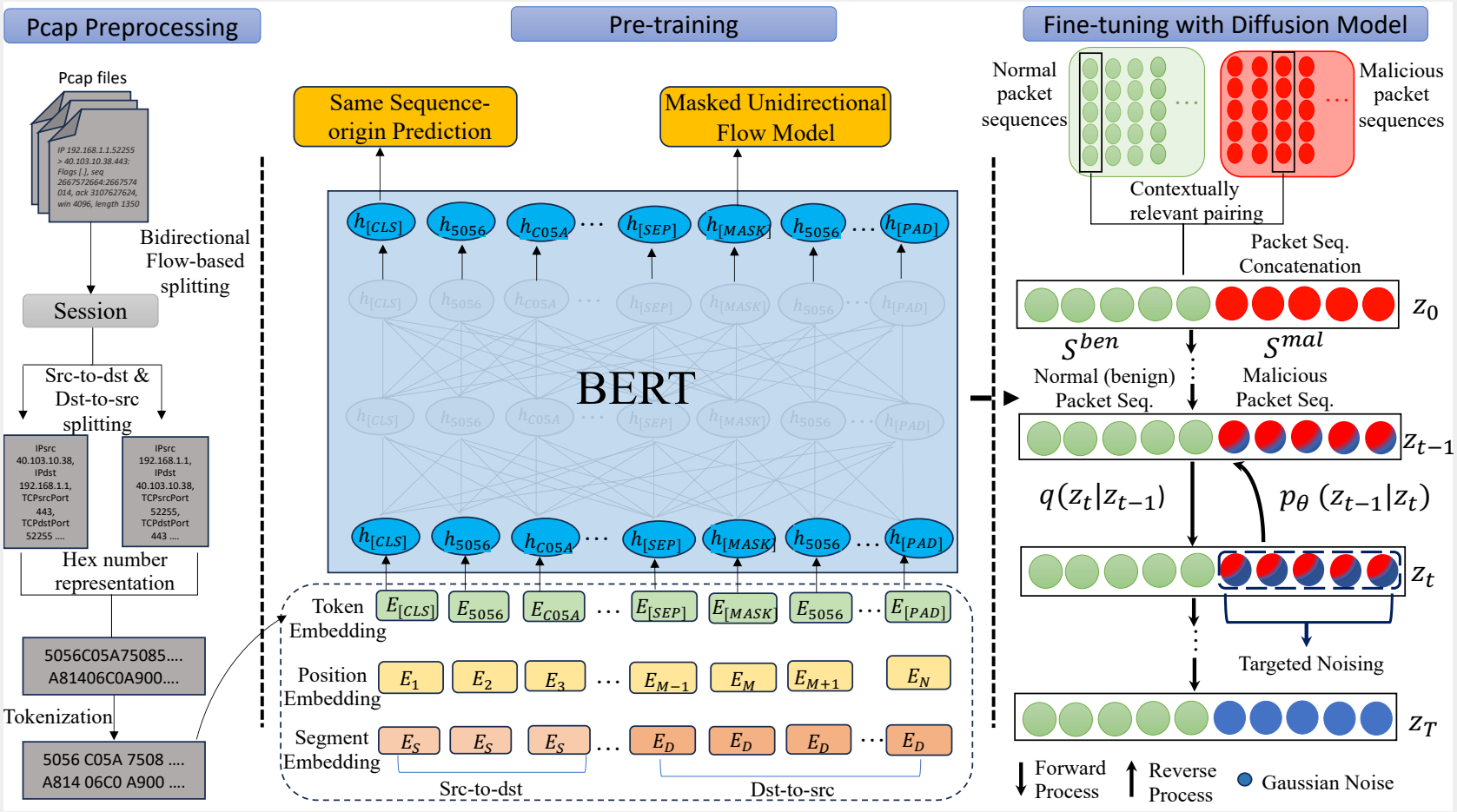


Fig. 3 Proposed Architecture

Fine-tuning with Diffusion Model

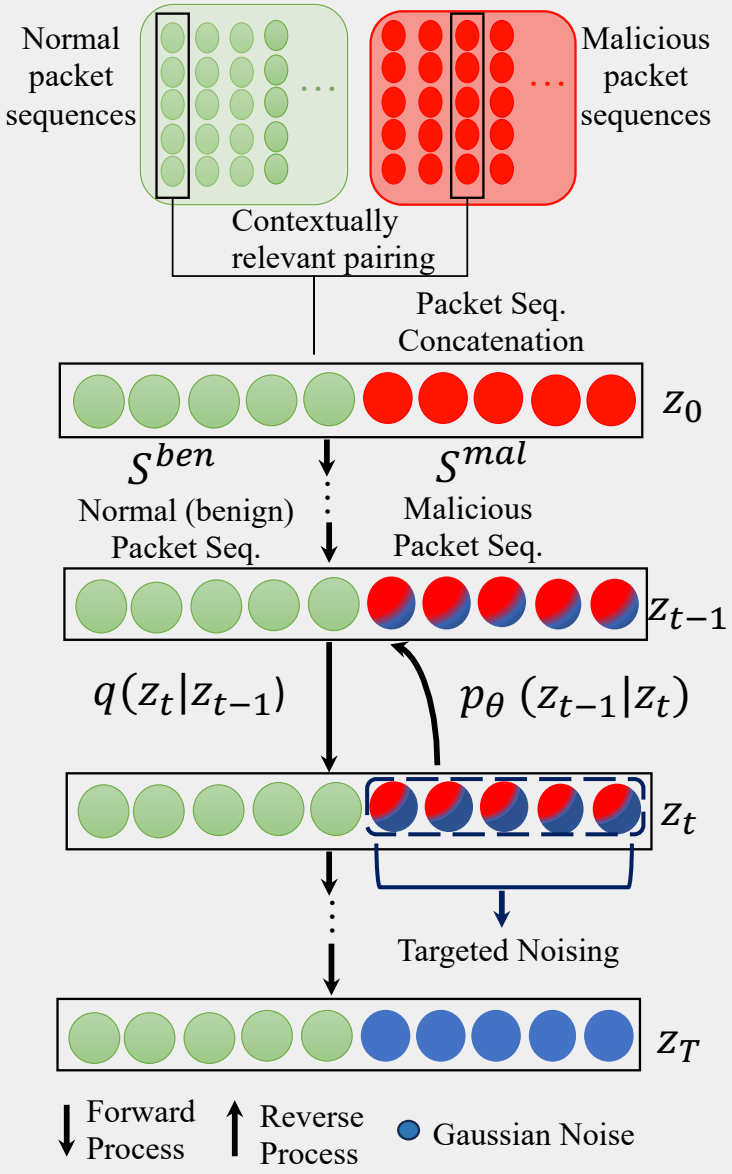


Fig. 4 Fine-tuning with Diffusion Model

- **Targeted Noising**
  - Inject noise into only the malicious packet sequences
- **Reverse Process**
  - **Guiding Framework:** Utilizes normal packet sequences as the guidance
    - Help the model to perceive the malicious elements as integral parts of the normal traffic pattern
  - **Denoising technique:** Treats the concatenated sequences of normal and noise-added malicious packets as a unified unit

MER Results

(a) Botnet

Feature Extractor	Classifier	Detection			Evasion Rate (*MER*)		
		P	R	F1	GAN & PSO	LSTM	**Ours**
CIC FLowMeter	KitNET	0.84	0.94	0.92	37.24%	**50.69%**	46.48%
	DT	0.79	0.91	0.82	35.88%	49.62%	**60.98%**
	IF	0.99	0.90	0.95	39.78%	42.93%	**49.71%**
	MLP	0.92	0.84	0.86	41.05%	53.87%	**63.05%**
	SVM	0.99	0.92	0.95	88.79%	**91.95%**	64.49%
	LR	0.84	0.91	0.89	24.72%	30.52%	**42.08%**
AfterImage	KitNET	0.96	0.90	0.94	99.18%	**99.79%**	74.46%
	DT	0.79	0.90	0.84	63.42%	67.30%	**72.13%**
	IF	0.99	0.90	0.94	31.48%	**61.08%**	52.79%
	MLP	0.96	0.97	0.97	48.60%	60.97%	**64.92%**
	SVM	0.99	0.90	0.94	40.31%	51.24%	**69.19%**
	LR	0.96	0.90	0.93	53.28%	50.70%	**58.98%**

(b) MITM

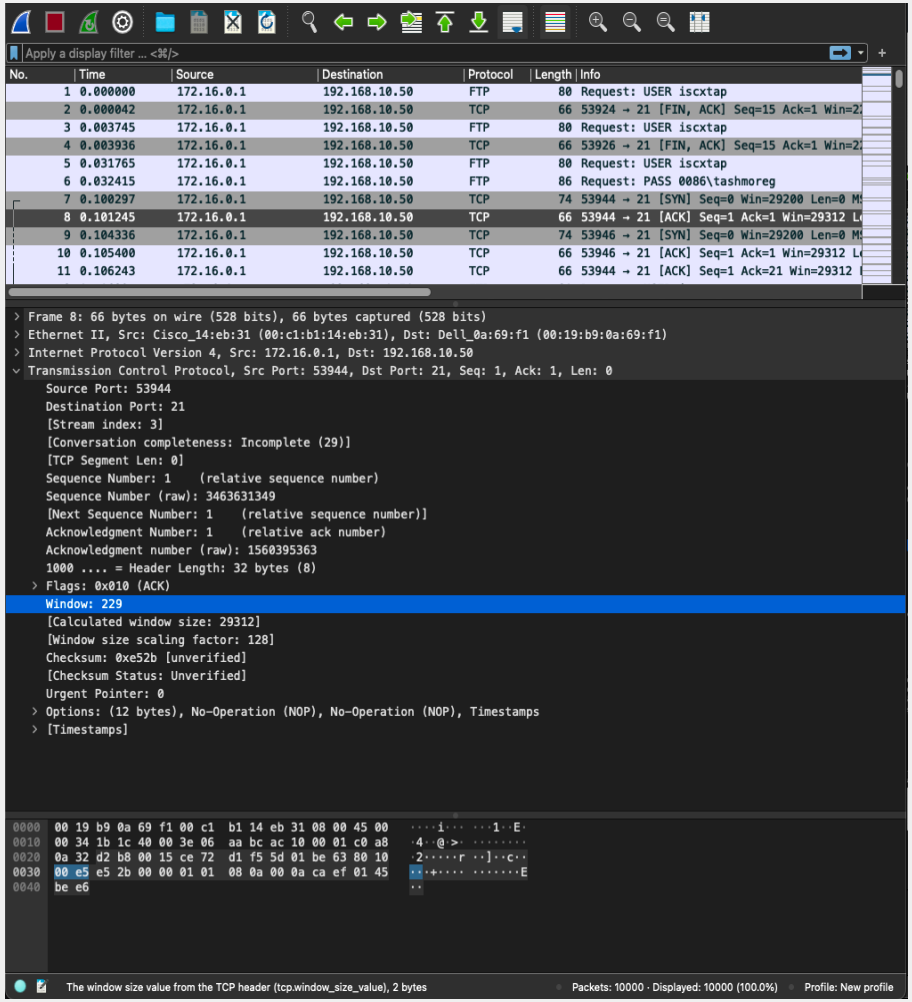
Feature Extractor	Classifier	Detection			Evasion Rate (*MER*)		
		P	R	F1	GAN & PSO	LSTM	**Ours**
CIC FLowMeter	KitNET	0.92	0.94	0.91	38.22%	44.39%	**53.87%**
	DT	0.74	0.79	0.76	49.98%	57.77%	**64.12%**
	IF	0.99	0.92	0.94	26.74%	38.64%	**52.99%**
	MLP	0.77	0.72	0.74	52.07%	43.19%	**73.21%**
	SVM	0.74	0.79	0.78	42.11%	45.54%	**60.46%**
	LR	0.73	0.78	0.72	35.87%	**50.04%**	47.48%
AfterImage	KitNET	0.94	0.96	0.93	68.79%	**79.29%**	58.48%
	DT	0.75	0.89	0.84	53.18%	58.15%	**70.04%**
	IF	0.81	0.83	0.86	26.53%	31.27%	**45.71%**
	MLP	0.92	0.90	0.93	50.65%	59.30%	**71.45%**
	SVM	0.99	0.90	0.94	63.51%	57.51%	**66.53%**
	LR	0.91	0.94	0.90	44.68%	46.39%	**52.38%**

Table 2: Comparative Analysis of Attack Detection and Evasion Rates

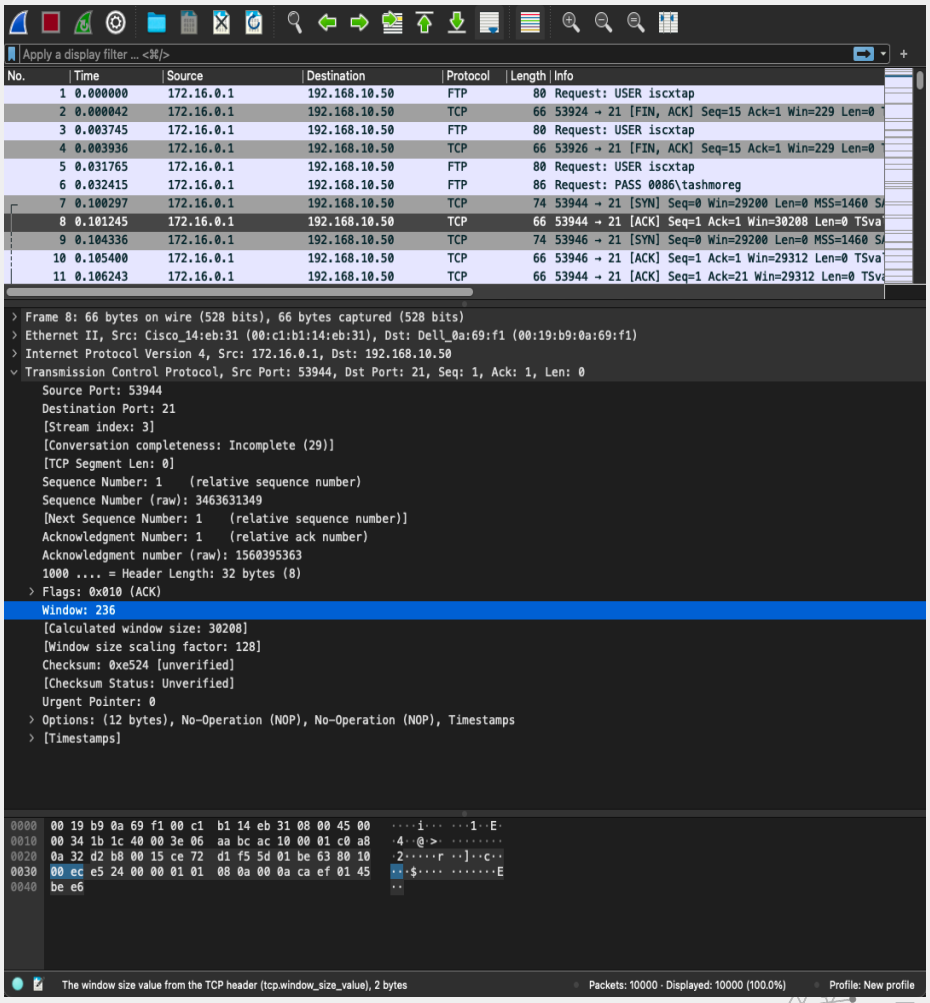
- **Malicious traffic Evasion Rate (MER) :** $\mathrm{MER} = 1 - (\mathrm{N}^{\mathrm{adv}}/\mathrm{N}^{\mathrm{mal}})$
- **Ours vs Traffic Manipulator :** ***9.12*** **percentage points on average**
- **Ours vs. TANTRA :** ***4.26*** **percentage points on average**
- **Overall Improvement :** ***6.69*** **percentage points higher**

Wireshark Botnet

- Botnet : Shows abnormal patterns in window size due to automated control and data bursts
- DiffuPac made the intelligent decision by itself in modifying the window size to fall within typical user traffic patterns



(a) Botnet attack before in Wireshark.



(b) Botnet attack after in Wireshark.

Figure 5: Comparison of Botnet attack before and after in Wireshark.

Malicious Functionality Evaluation (Brute Force)

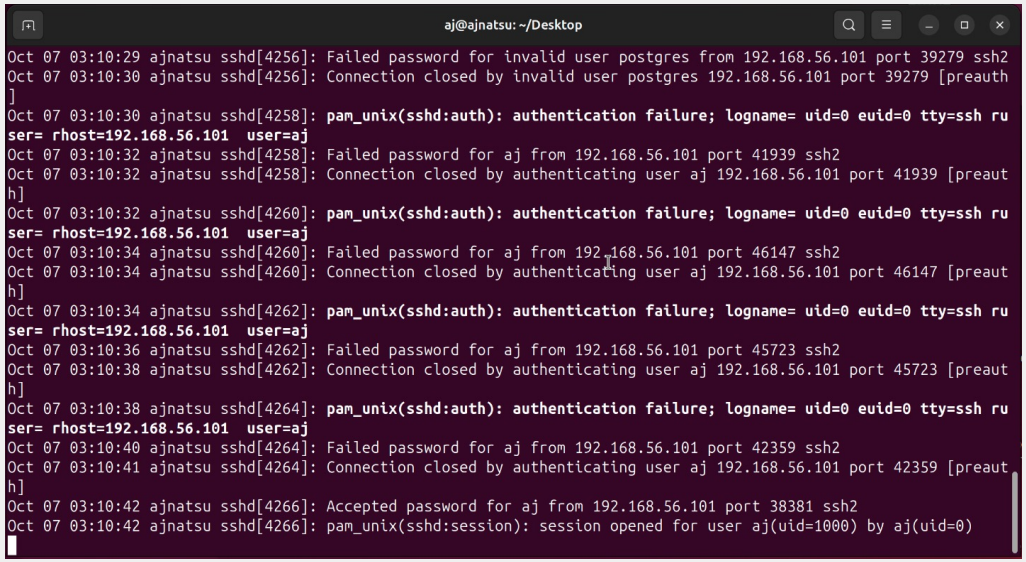


Fig.10 Original Brute Force Attack response

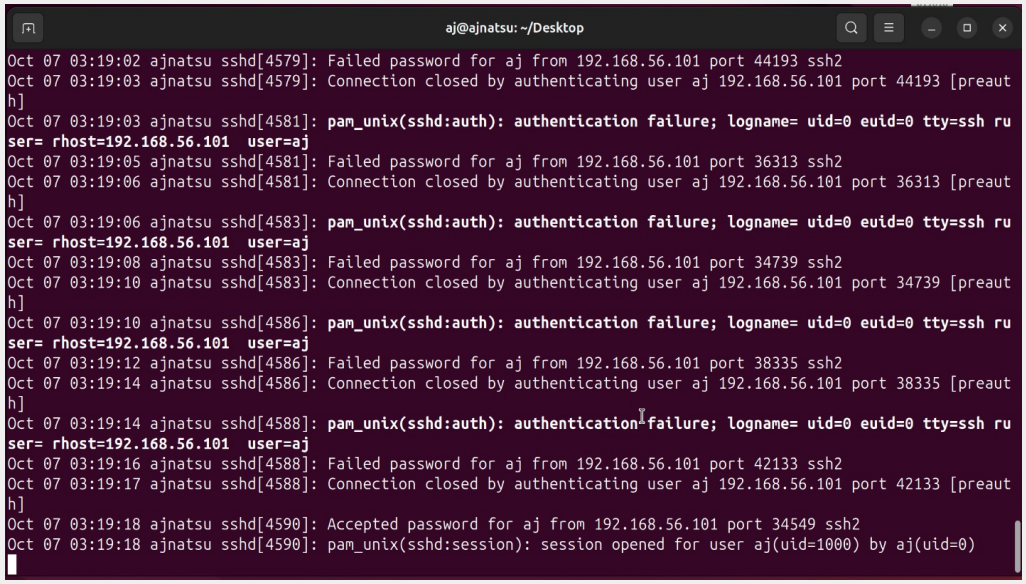


Fig.11 Adversarial Brute Force Attack response

- **Focused on analyzing the response of both original and adversarial packets**
- **Environment 1 server -to- 1 client: Kali Linux (Attacker) -to- Ubuntu (Victim)**
- **Log entries confirmed that the SSH service recorded successful logins for both the original Brute Force attack and the adversarial Brute Force attack**