

# 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

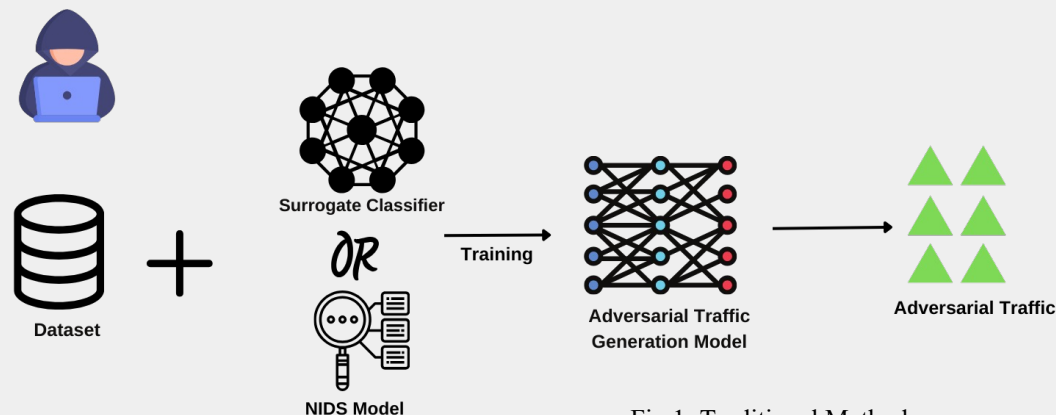


Fig.1: Traditional Methods

# 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

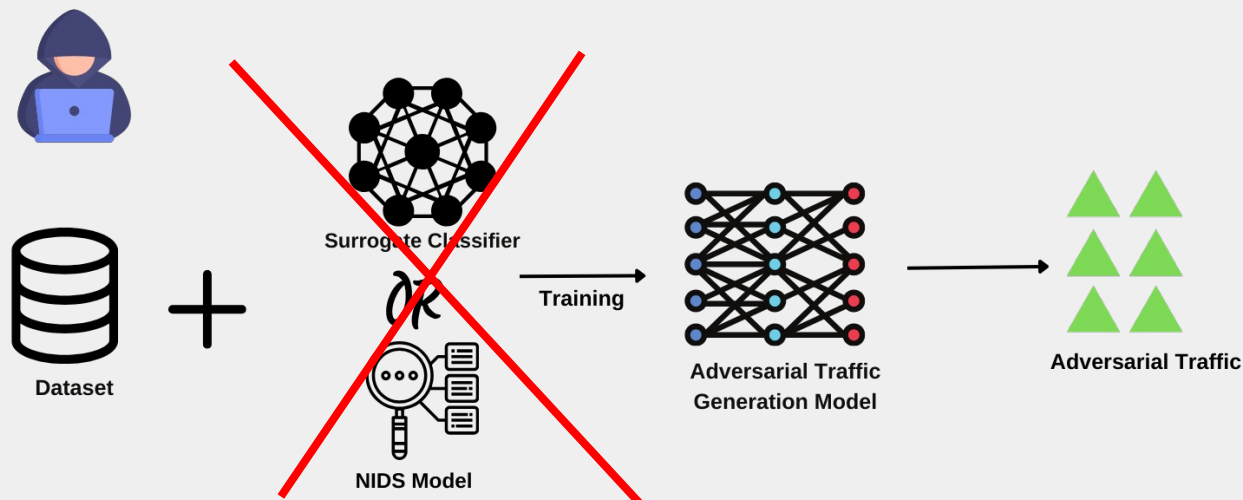


Fig. 2: New Method of Adversarial Generation

# Proposed Architecture

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

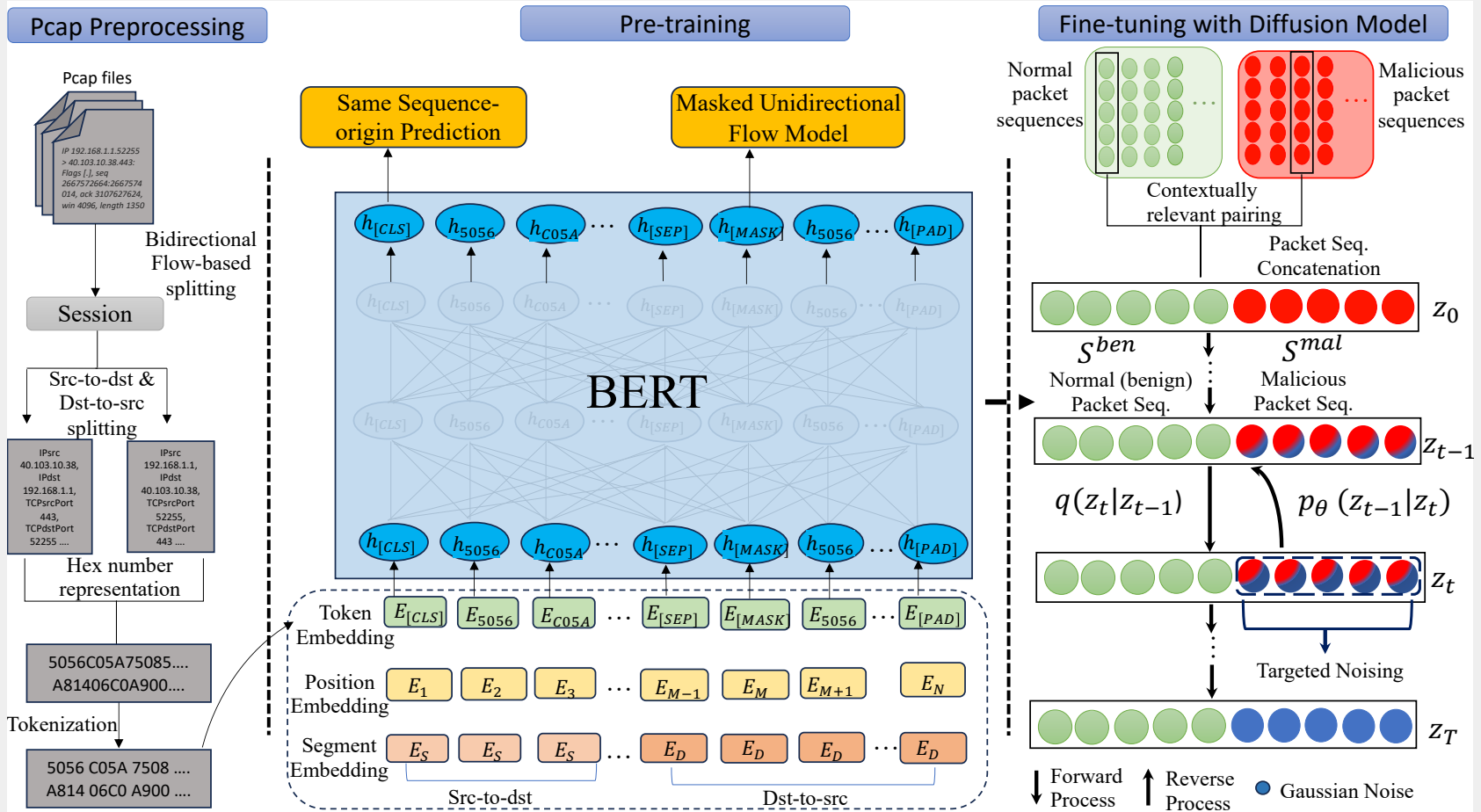


Fig. 3 Proposed Architecture

# Fine-tuning with Diffusion Model

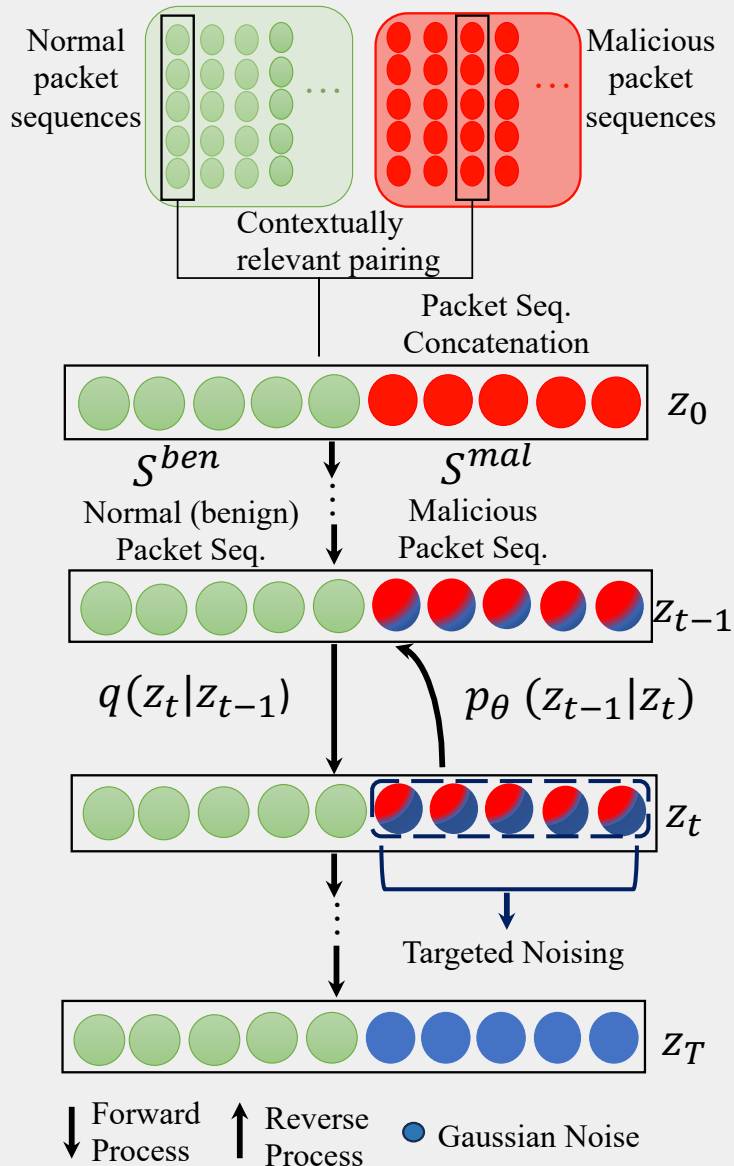


Fig. 4 Fine-tuning with Diffusion Model

## ◆ 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
- ◆ **Denosing 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%	<b>50.69%</b>	46.48%
	DT	0.79	0.91	0.82	35.88%	49.62%	<b>60.98%</b>
	IF	0.99	0.90	0.95	39.78%	42.93%	<b>49.71%</b>
	MLP	0.92	0.84	0.86	41.05%	53.87%	<b>63.05%</b>
	SVM	0.99	0.92	0.95	88.79%	<b>91.95%</b>	64.49%
	LR	0.84	0.91	0.89	24.72%	30.52%	<b>42.08%</b>
AfterImage	KitNET	0.96	0.90	0.94	99.18%	<b>99.79%</b>	74.46%
	DT	0.79	0.90	0.84	63.42%	67.30%	<b>72.13%</b>
	IF	0.99	0.90	0.94	31.48%	<b>61.08%</b>	52.79%
	MLP	0.96	0.97	0.97	48.60%	60.97%	<b>64.92%</b>
	SVM	0.99	0.90	0.94	40.31%	51.24%	<b>69.19%</b>
	LR	0.96	0.90	0.93	53.28%	50.70%	<b>58.98%</b>

		(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%	<b>53.87%</b>
	DT	0.74	0.79	0.76	49.98%	57.77%	<b>64.12%</b>
	IF	0.99	0.92	0.94	26.74%	38.64%	<b>52.99%</b>
	MLP	0.77	0.72	0.74	52.07%	43.19%	<b>73.21%</b>
	SVM	0.74	0.79	0.78	42.11%	45.54%	<b>60.46%</b>
	LR	0.73	0.78	0.72	35.87%	<b>50.04%</b>	47.48%
AfterImage	KitNET	0.94	0.96	0.93	68.79%	<b>79.29%</b>	58.48%
	DT	0.75	0.89	0.84	53.18%	58.15%	<b>70.04%</b>
	IF	0.81	0.83	0.86	26.53%	31.27%	<b>45.71%</b>
	MLP	0.92	0.90	0.93	50.65%	59.30%	<b>71.45%</b>
	SVM	0.99	0.90	0.94	63.51%	57.51%	<b>66.53%</b>
	LR	0.91	0.94	0.90	44.68%	46.39%	<b>52.38%</b>

- ◆ Malicious traffic Evasion Rate (MER) :  $MER = 1 - (N^{adv}/N^{mal})$
- ◆ 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

Table 2: Comparative Analysis of Attack Detection and Evasion Rates

# 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

Apply a display filter ... <#>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
2	0.000042	172.16.0.1	192.168.10.50	TCP	66	53924 → 21 [FIN, ACK] Seq=15 Ack=1 Win=229 Len=0
3	0.003745	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
4	0.003936	172.16.0.1	192.168.10.50	TCP	66	53926 → 21 [FIN, ACK] Seq=15 Ack=1 Win=229 Len=0
5	0.031765	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
6	0.032415	172.16.0.1	192.168.10.50	FTP	86	Request: PASS 0086\tashmoreg
7	0.100297	172.16.0.1	192.168.10.50	TCP	74	53944 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 S
8	0.101245	172.16.0.1	192.168.10.50	TCP	66	53944 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0
9	0.104336	172.16.0.1	192.168.10.50	TCP	74	53946 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 S
10	0.105400	172.16.0.1	192.168.10.50	TCP	66	53946 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0
11	0.106243	172.16.0.1	192.168.10.50	TCP	66	53944 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0

Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface II, Src: Cisco\_14:eb:31 (00:c1:b1:14:eb:31), Dst: Dell\_0a:69:f1 (00:19:b9:0a:69:f1)

Internet Protocol Version 4, Src: 172.16.0.1, Dst: 192.168.10.50

Transmission Control Protocol, Src Port: 53944, Dst Port: 21, Seq: 1, Ack: 1, Len: 0

Source Port: 53944  
Destination Port: 21  
[Stream index: 3]  
[Conversation completeness: Incomplete (29)]  
[TCP Segment Len: 0]  
Sequence Number: 1 (relative sequence number)  
Sequence Number (raw): 3463631349  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment number (raw): 1560395363  
1000 .... = Header Length: 32 bytes (8)  
> Flags: 0x010 (ACK)  
**Window: 229**  
[Calculated window size: 29312]  
[Window size scaling factor: 128]  
Checksum: 0xe52b [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps  
> [Timestamps]

0000 00 19 b9 0a 69 f1 00 c1 b1 14 eb 31 08 00 45 00 ...1...1..E.  
0010 00 34 1b 1c 40 00 3e 06 aa bc ac 10 00 01 c0 a8 ..4..@>.....  
0020 0a 32 d2 b8 00 15 ce 72 d1 f5 5d 01 be 63 80 10 ..2.....F...c...  
0030 00 e5 e5 2b 00 00 01 01 08 0a 00 0a ca ef 01 45 ..\$......E  
0040 be e6 ..

The window size value from the TCP header (tcp.window\_size\_value), 2 bytes      Packets: 10000 · Displayed: 10000 (100.0%)      Profile: New profile

(a) Botnet attack before in Wireshark.

Apply a display filter ... <#>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
2	0.000042	172.16.0.1	192.168.10.50	TCP	66	53924 → 21 [FIN, ACK] Seq=15 Ack=1 Win=229 Len=0
3	0.003745	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
4	0.003936	172.16.0.1	192.168.10.50	TCP	66	53926 → 21 [FIN, ACK] Seq=15 Ack=1 Win=229 Len=0
5	0.031765	172.16.0.1	192.168.10.50	FTP	80	Request: USER iscsxtap
6	0.032415	172.16.0.1	192.168.10.50	FTP	86	Request: PASS 0086\tashmoreg
7	0.100297	172.16.0.1	192.168.10.50	TCP	74	53944 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 S
8	0.101245	172.16.0.1	192.168.10.50	TCP	66	53944 → 21 [ACK] Seq=1 Ack=1 Win=30208 Len=0 TSva
9	0.104336	172.16.0.1	192.168.10.50	TCP	74	53946 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 S
10	0.105400	172.16.0.1	192.168.10.50	TCP	66	53946 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSva
11	0.106243	172.16.0.1	192.168.10.50	TCP	66	53944 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0 TSva

Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface II, Src: Cisco\_14:eb:31 (00:c1:b1:14:eb:31), Dst: Dell\_0a:69:f1 (00:19:b9:0a:69:f1)

Internet Protocol Version 4, Src: 172.16.0.1, Dst: 192.168.10.50

Transmission Control Protocol, Src Port: 53944, Dst Port: 21, Seq: 1, Ack: 1, Len: 0

Source Port: 53944  
Destination Port: 21  
[Stream index: 3]  
[Conversation completeness: Incomplete (29)]  
[TCP Segment Len: 0]  
Sequence Number: 1 (relative sequence number)  
Sequence Number (raw): 3463631349  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment number (raw): 1560395363  
1000 .... = Header Length: 32 bytes (8)  
> Flags: 0x010 (ACK)  
**Window: 236**  
[Calculated window size: 30208]  
[Window size scaling factor: 128]  
Checksum: 0xe524 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps  
> [Timestamps]

0000 00 19 b9 0a 69 f1 00 c1 b1 14 eb 31 08 00 45 00 ...1...1..E.  
0010 00 34 1b 1c 40 00 3e 06 aa bc ac 10 00 01 c0 a8 ..4..@>.....  
0020 0a 32 d2 b8 00 15 ce 72 d1 f5 5d 01 be 63 80 10 ..2.....F...c...  
0030 00 ec e5 24 00 00 01 01 08 0a 00 0a ca ef 01 45 ..\$......E  
0040 be e6 ..

The window size value from the TCP header (tcp.window\_size\_value), 2 bytes      Packets: 10000 · Displayed: 10000 (100.0%)      Profile: New profile

(b) Botnet attack after in Wireshark.

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



# Malicious Functionality Evaluation (Brute Force)

```

aj@ajnatsu: ~/Desktop
Oct 07 03:10:29 ajnatsu sshd[4256]: Failed password for invalid user postgres from 192.168.56.101 port 39279 ssh2
Oct 07 03:10:30 ajnatsu sshd[4256]: Connection closed by invalid user postgres 192.168.56.101 port 39279 [preauth
]
Oct 07 03:10:30 ajnatsu sshd[4258]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:10:32 ajnatsu sshd[4258]: Failed password for aj from 192.168.56.101 port 41939 ssh2
Oct 07 03:10:32 ajnatsu sshd[4258]: Connection closed by authenticating user aj 192.168.56.101 port 41939 [preaut
h]
Oct 07 03:10:32 ajnatsu sshd[4260]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:10:34 ajnatsu sshd[4260]: Failed password for aj from 192.168.56.101 port 46147 ssh2
Oct 07 03:10:34 ajnatsu sshd[4260]: Connection closed by authenticating user aj 192.168.56.101 port 46147 [preaut
h]
Oct 07 03:10:34 ajnatsu sshd[4262]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:10:36 ajnatsu sshd[4262]: Failed password for aj from 192.168.56.101 port 45723 ssh2
Oct 07 03:10:38 ajnatsu sshd[4262]: Connection closed by authenticating user aj 192.168.56.101 port 45723 [preaut
h]
Oct 07 03:10:38 ajnatsu sshd[4264]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:10:40 ajnatsu sshd[4264]: Failed password for aj from 192.168.56.101 port 42359 ssh2
Oct 07 03:10:41 ajnatsu sshd[4264]: Connection closed by authenticating user aj 192.168.56.101 port 42359 [preaut
h]
Oct 07 03:10:42 ajnatsu sshd[4266]: Accepted password for aj from 192.168.56.101 port 38381 ssh2
Oct 07 03:10:42 ajnatsu sshd[4266]: pam_unix(sshd:session): session opened for user aj(uid=1000) by aj(uid=0)

```

Fig.10 Original Brute Force Attack response

```

aj@ajnatsu: ~/Desktop
Oct 07 03:19:02 ajnatsu sshd[4579]: Failed password for aj from 192.168.56.101 port 44193 ssh2
Oct 07 03:19:03 ajnatsu sshd[4579]: Connection closed by authenticating user aj 192.168.56.101 port 44193 [preaut
h]
Oct 07 03:19:03 ajnatsu sshd[4581]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:19:05 ajnatsu sshd[4581]: Failed password for aj from 192.168.56.101 port 36313 ssh2
Oct 07 03:19:06 ajnatsu sshd[4581]: Connection closed by authenticating user aj 192.168.56.101 port 36313 [preaut
h]
Oct 07 03:19:06 ajnatsu sshd[4583]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:19:08 ajnatsu sshd[4583]: Failed password for aj from 192.168.56.101 port 34739 ssh2
Oct 07 03:19:10 ajnatsu sshd[4583]: Connection closed by authenticating user aj 192.168.56.101 port 34739 [preaut
h]
Oct 07 03:19:10 ajnatsu sshd[4586]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:19:12 ajnatsu sshd[4586]: Failed password for aj from 192.168.56.101 port 38335 ssh2
Oct 07 03:19:14 ajnatsu sshd[4586]: Connection closed by authenticating user aj 192.168.56.101 port 38335 [preaut
h]
Oct 07 03:19:14 ajnatsu sshd[4588]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ru
ser= rhost=192.168.56.101 user=aj
Oct 07 03:19:16 ajnatsu sshd[4588]: Failed password for aj from 192.168.56.101 port 42133 ssh2
Oct 07 03:19:17 ajnatsu sshd[4588]: Connection closed by authenticating user aj 192.168.56.101 port 42133 [preaut
h]
Oct 07 03:19:18 ajnatsu sshd[4590]: Accepted password for aj from 192.168.56.101 port 34549 ssh2
Oct 07 03:19:18 ajnatsu sshd[4590]: pam_unix(sshd:session): session opened for user aj(uid=1000) by aj(uid=0)

```

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