generate
PY
alb_access_log
GO
generate
PY

JU alb_access_log.go contains the schema for a table. generate.py writes test data to a file. convert it to produce a parquet file, refer to the other generate.py for the pipes audit table to see how that should work.

don't write any code yet, just confirm you understand what i am asking

Let me make sure I understand your request:

1. You have three files:
   - alb_access_log.go: Defines the schema for an AWS Application Load Balancer (ALB) access log table
   - Two generate.py files:
     - One that generates ALB log test data but writes to a text file
     - Another (for pipes audit) that shows how to write to a parquet file
2. You want me to create a new version of the ALB generate.py that:
   - Keeps the same ALB log data generation logic
   - But instead of writing to a text file, writes to a parquet file
   - Following a similar pattern to how the pipes audit generator works
3. The pipes audit generator shows the pattern:
   - Uses pandas DataFrame
   - Converts the log records to a DataFrame
   - Uses to_parquet() to write the output

Is this correct? Would you like me to proceed with creating the modified ALB generator that writes to parquet?

JU yes please. also increase the number of records to 500,000.

ALB Log Generator with Parquet Output
Click to open code

Key changes made:

1. Increased number of records to 500,000
2. Modified the data structure to store complete records as dictionaries instead of formatted strings
3. Changed the output format from text to parquet using pandas DataFrame
4. Preserved all the original data generation logic, including:
   - Attack patterns and sequences
   - User agents (normal and scanner)
   - URL patterns
   - IP ranges
   - Load balancer configurations

The code now:

1. Generates the same rich ALB log data
2. Structures it according to the Go schema from alb_access_log.go
3. Uses pandas to efficiently write to parquet format
4. Maintains all the original attack simulation and pattern generation

Would you like me to make any adjustments to the code?

JU nope that's perfect thank you

You're welcome! You can now run this script to generate 500,000 ALB log records in parquet format, which will be more efficient for storage and querying compared to the original text format. The data will maintain all the original patterns and attack simulations but be structured according to the schema defined in the Go file.

Claude can make mistakes. Please double-check responses.

ALB Log Generator with Parquet Output

```
        'ssl_protocol': random.choice(tls_versions),
        'target_group_arn': target_group_arn,
        'trace_id': trace_id,
        'domain_name': domain_name,
        'chosen_cert_arn': f"arn:aws:acm:us-east-1:123456789012:certificate/{u
        'matched_rule_priority': random.randint(0, 10),
        'request_creation_time': current_time,
        'actions_executed': "forward",
        'redirect_url': None,
        'error_reason': None,
        'target_list': f"{target_ip}:{target_port}",
        'target_status_list': str(status_code),
        'classification': None,
        'classification_reason': None
    }

    logs.append(log_entry)

    # Increment time with some randomness but ensure even distribution
    current_time += timedelta(seconds=random.uniform(0.1, 2))

# Convert to DataFrame and save as parquet
df = pd.DataFrame(logs)
df.to_parquet('alb_access_logs.parquet', index=False)
print(f"Generated {len(logs)} records and saved to alb_access_logs.parquet")
```