Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning.

```
# Define initial facts and rules
facts = {"InAmerica(West)", "SoldWeapons(West, Nono)", "Enemy(Nono, America)"}
rules = [
    {
        "conditions": ["InAmerica(x)", "SoldWeapons(x, y)", "Enemy(y, America)"],
        "conclusion": "Criminal(x)",
    },
    {
        "conditions": ["Enemy(y, America)"],
        "conclusion": "Dangerous(y)",
    },
]

# Forward chaining function
def forward_chaining(facts, rules):
    derived_facts = set(facts)  # Initialize derived facts
    while True:
        new_fact_found = False

        for rule in rules:
            # Substitute variables and check if conditions are met
            for fact in derived_facts:
                if "x" in rule["conditions"][0]:
                    # Substitute variables (x, y) with specific instances
                    for condition in rule["conditions"]:
                        if "x" in condition or "y" in condition:
                            x = "West"  # Hardcoded substitution for simplicity
                            y = "Nono"
                            conditions = [
                                cond.replace("x", x).replace("y", y)
                                for cond in rule["conditions"]
                            ]
                            conclusion = (
                                rule["conclusion"].replace("x", x).replace("y", y)
                            )

                            # Check if all conditions are satisfied
                            if all(cond in derived_facts for cond in conditions) and conclusion not in derived_facts:
                                derived_facts.add(conclusion)
                                print(f"New fact derived: {conclusion}")
                                new_fact_found = True

        # Exit loop if no new fact is found
        if not new_fact_found:
            break

    return derived_facts

# Run forward chaining
final_facts = forward_chaining(facts, rules)
print("Output: 1BM22CS290")
print("\nFinal derived facts:")
for fact in final_facts:
    print(fact)
```

```
Output: 1BM22CS290

Final derived facts:
SoldWeapons(West, Nono)
InAmerica(West)
Enemy(Nono, America)
```