ارامكو السعودية
Saudi Aramco

Techniques & Algorithms for ACL Optimization

Ibrahim Al Abdulmohsin

Communications Eng. & Technical Support Dept.

Saudi Aramco

OBJECTIVE

To explain the problems associated with access control lists, and how simple optimization techniques can alleviate such problems significantly

AGENDA

- Background
- ACL Definition and Problem Statement
- Optimization Techniques & Algorithms
- ACLO Application Overview
- Results and Conclusions

ارامكو السعودية
Saudi Aramco

Networking Introduced

Data Networking

is the science of linking together a group of two or more devices to allow for transmission of data and sharing of resources.

- **Devices :** computers, telephones, hardware appliances, network elements ..etc
- **Transmission :** unidirectional, bidirectional, one-to-one, one-to-many ..etc

ارامكو السعودية
Saudi Aramco

Background

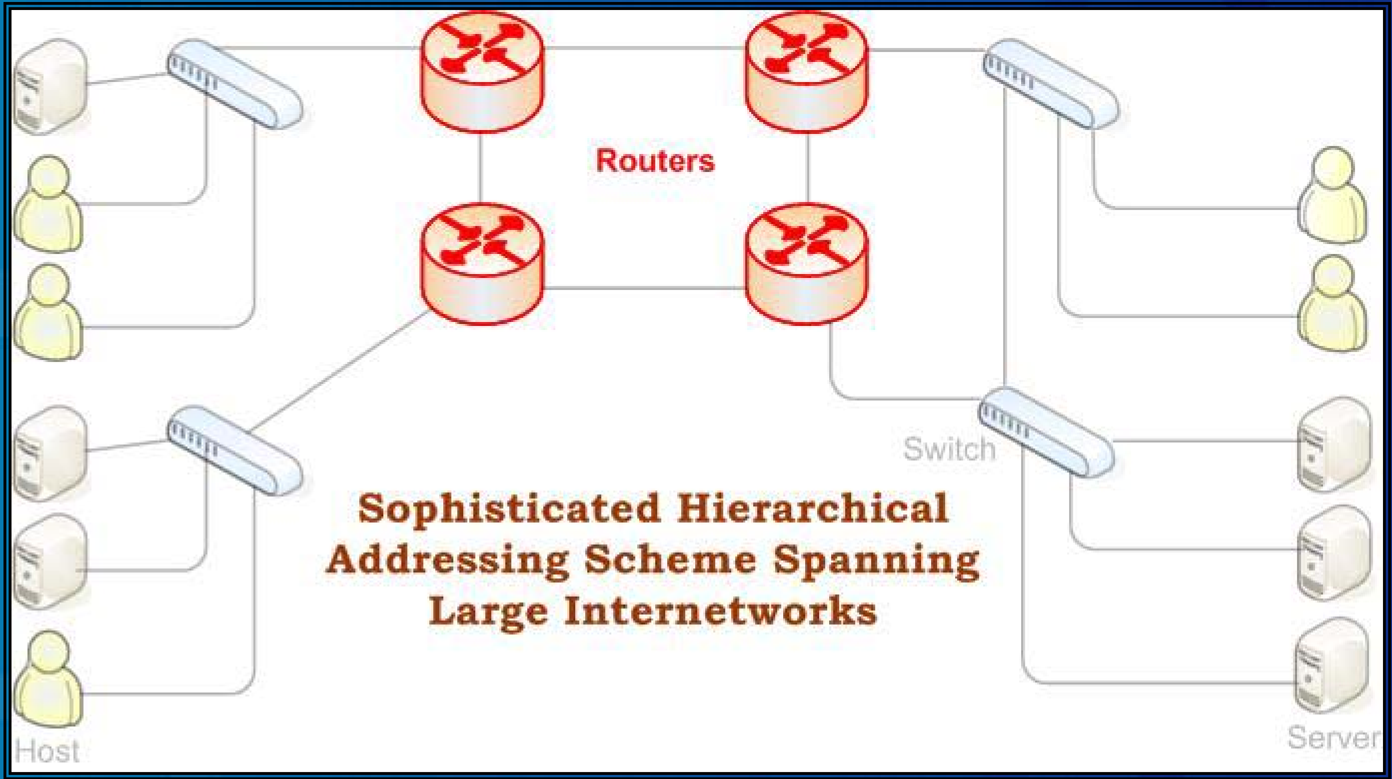
Networking Introduced

OSI Reference Model

Seven Layers:

1. **PHYSICAL**
2. **DATA LINK**
3. **NETWORK**
4. **TRANSPORT**
5. **SESSION**
6. **PRESENTATION**
7. **APPLICATION**

Layer 3: Network Layer



- **Routed Protocols such as IP and IPX, and routing protocols such as OSPF, IS-IS, and BGP are L3 protocols.**

Networking Evolved

1st Phase:

- Technologies that facilitate exchange of information
  - Routing protocols, fast Layer2 technologies, ...etc

2nd Phase:

- Technologies that are byproduct of networking
  - Network Security, Network Management Systems, ...etc

3rd Phase:

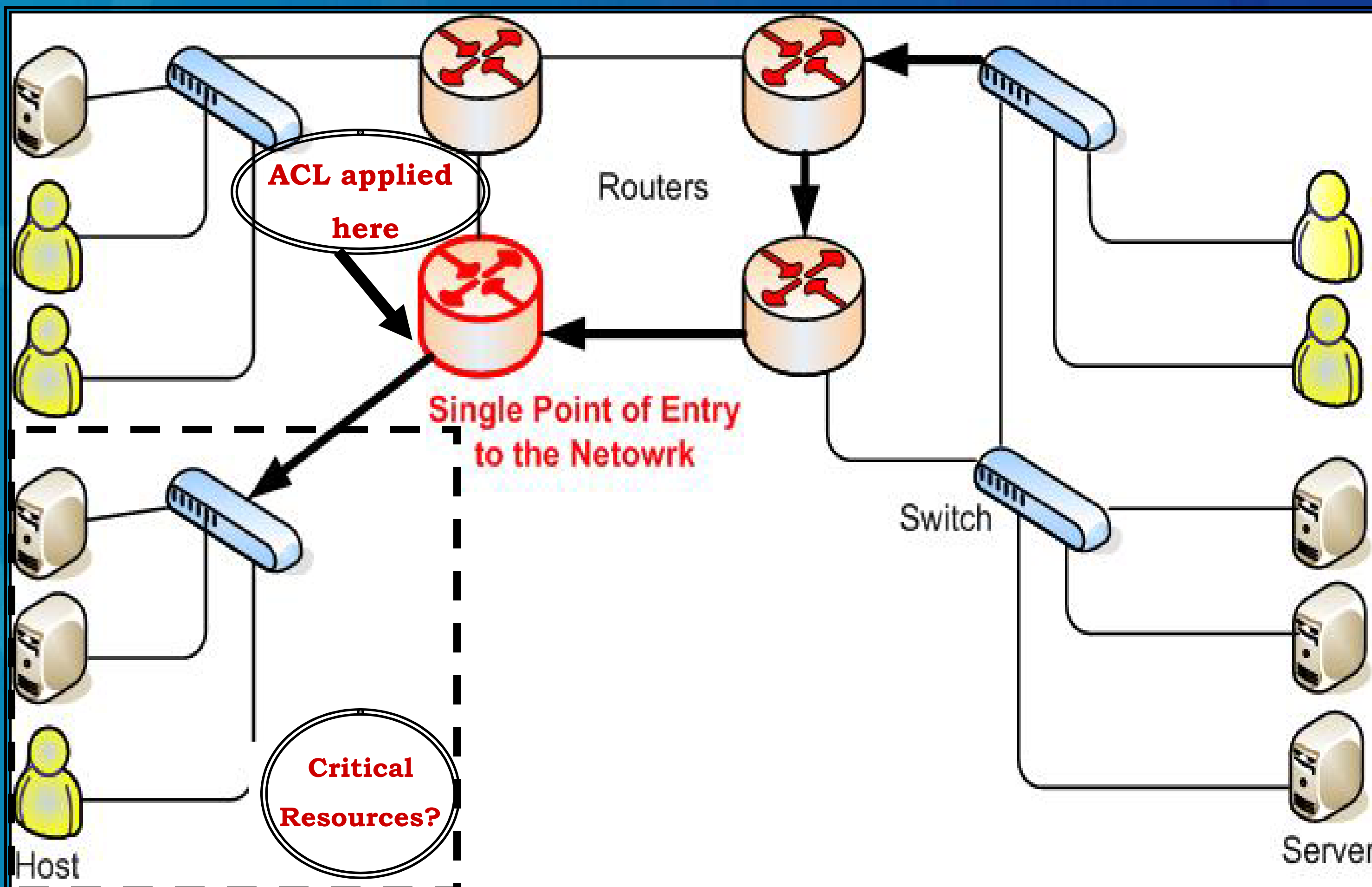
- Technologies that enhance the operation of existing infrastructures
  - SANs, Content Networking, ...etc

4th Phase:

- Convergence
  - Voice over IP, Fax over IP, Video over IP, ...etc

Role of Routers in Today's Networks

4. Control Over Traffic Flow: QoS, ACL



What is an ACL?

- **ACL (Access Control List):** A policy that determines which traffic should be permitted and which should be denied.

Packet p = {icmp, 10.1.12.3 , 10.114.48.228 , echo-reply}

- **Rule:** A 6-tuple object = {action, protocol, source address range, source port range, destination address range, destination port range}

```
Permit tcp host 10.1.182.184  eq 1520  10.201.165.0 0.0.0.255  eq 1521
```

- An ACL is composed of multiple rules, executed sequentially

p ✗ ✓

```
Permit udp   host 10.1.12.2        any                          eq snmp
Permit ip    10.1.12.2  0.0.0.3    any
Permit ip    host 10.1.12.3        any
Permit icmp  host 10.114.1.100     host 10.114.48.228           echo
Permit icmp  host 10.114.1.100     host 10.114.48.228           echo-reply
permit tcp   any                   10.114.48.0  0.0.0.255  (95 matches)
deny   ip    any                   any  (365144 matches)
```

Saudi Aramco

Quick Definitions

- **Executed Rule:** ***Ri*** ε ***pk*****, read** ***Ri*** **executed for the** ***k*****th packet → action field in** ***Ri*** **determines whether the packet** ***pk*** **will be permitted or denied.**
- **Hits Probability of Rule** ***Ri*** **(** ***h(i)*** **):**

$$h(i) = matches\ (i) / \sum_{j=1}^{ACL\ size} match\ (j)$$

- **Rule Packet Latenc**

$$RPL(i) = \sum_{0}^{i} RL(k)$$

- **Expected Packet L**

$$EPL = \sum_{0}^{n-1} h(i) RPL(i)$$

Problem Statement

Delay

- Addition of ACLs increases end-to-end packet latency.
- Delay for a single packet depends on the location of the executed rule.

P1 = {icmp, 10.1.12.3 , 10.114.48.228 , echo-reply)

p1 p2 P

X	Permit	udp	host 10.1.12.2	any	eq snmp
X	Permit	ip	10.1.12.2 0.0.0.3	any	
X	Deny	ip	host 10.1.12.3	any	
X	Permit	icmp	host 10.114.1.100	host 10.114.48.228	echo
X	Permit	icmp	host 10.114.1.100	host 10.114.48.228	echo-reply
√	permit	tcp	any	10.114.48.0 0.0.0.255	(95 matches)
	deny	ip	any	any (365144 matches)	

Increase in Packet latency for p2 = six rule latencies

Saudi Aramco
ارامكو السعودية

Problem Statement

Configuration Errors

- Rule 3 is

Permit udp	hos			eq snmp
Permit ip	10.			
Deny ip	hos			
Permit icmp	hos		.228	echo
Permit icmp	hos		.228	echo-reply
permit ip	an		.0.0.255	(95 matches)
deny ip	any		atches)	

```
permit icmp host 10.114.1.100 host 10.114.48.228 echo
permit icmp host 10.114.1.100 host 10.114.48.228 echo-reply
permit icmp host 1.1.1.100 host 111.114.48.227 echo
permit icmp host 1.1.1.100 host 111.114.48.227 echo-reply
permit icmp host 1.1.1.100 host 111.114.48.228 echo
permit icmp host 1.1.1.100 host 111.114.48.228 echo-reply
permit icmp host 1.1.1.100 host 111.114.49.227 echo
permit icmp host 1.1.1.100 host 111.114.49.227 echo-reply
permit icmp host 1.1.1.100 host 111.114.48.224 echo
permit icmp host 1.1.1.100 host 111.114.48.224 echo-reply
permit icmp host 1.1.1.100 host 111.114.48.228 echo
permit icmp host 1.1.1.100 host 111.114.48.228 echo-reply
permit icmp host 1.1.1.102 host 111.114.48.227 echo
permit tcp  host 1.1.1.7   eq 5450 10.114.8.253 0.0.3.0
permit tcp  host 1.1.1.221 eq 5450 10.114.8.253 0.0.3.0
permit tcp  host 1.1.1.182 eq 5450 10.114.8.253 0.0.3.0
permit tcp  host 1.1.1.224 10.114.8.253 0.0.3.0 eq ftp
permit tcp  host 1.1.1.253 0.0.3.0 eq ftp-data
permit tcp  host 1.1.1.224 10.114.8.253 0.0.3.0 eq telnet
permit icmp host 1.1.1.102 host 111.114.48.224 echo-reply
permit icmp host 1.1.1.103 host 111.114.48.228 echo
permit icmp host 1.1.1.103 host 111.114.48.228 echo-reply
permit icmp host 1.1.1.103 host 111.114.48.227 echo
permit icmp host 1.1.1.103 host 111.114.48.227 echo-reply
permit icmp host 1.1.1.103 host 111.114.49.228 echo
permit icmp host 1.1.1.103 host 111.114.49.228 echo-reply
permit icmp host 1.1.1.103 host 111.114.49.227 echo
permit icmp host 1.1.1.103 host 111.114.49.227 echo-reply
permit icmp host 1.1.1.103 host 111.114.48.224 echo
permit icmp host 1.1.1.103 host 111.114.48.224 echo-reply
permit tcp host 10.114.1.100 eq 5450 host 10.114.48.227
permit tcp host 10.114.1.100 eq 5450 host 10.114.49.227
permit tcp host 10.114.1.100 eq 5450 host 10.114.48.228
permit tcp host 10.114.1.100 eq 5450 host 10.114.49.228
permit tcp any 10.114.48.0 0.0.0.255 eq 5450
permit tcp any 10.114.48.0 0.0.0.255 eq 5631
permit tcp any 10.114.48.0 0.0.0.255 eq www
permit tcp any 10.114.49.0 0.0.0.255 eq www
permit udp host 10.1.161.32 any eq snmp
permit tcp any 2.2.48.0  0.0.0.255 eq 522
permit tcp any 2.2.48.0  0.0.0.255 eq 1503
permit tcp any 3.3.49.0  0.0.0.255 eq 522
permit tcp any 3.3.49.0  0.0.0.255 eq 1503
permit udp any 2.2.48.0  0.0.0.255 eq 5632
permit tcp any 3.3.49.0  0.0.0.255 eq 5631
permit udp any 3.3.49.0  0.0.0.255 eq 5632
permit tcp any 2.2.48.0  0.0.0.255 eq 13782
permit tcp any 2.2.48.0  0.0.0.255 eq 13720
permit tcp any 2.2.48.0  0.0.0.255 eq 13724
permit tcp any 3.3.49.0  0.0.0.255 eq 5450
permit tcp any 3.3.49.0  0.0.0.255 eq 13782
permit tcp any 3.3.49.0  0.0.0.255 eq 13720
permit tcp any 3.3.49.0  0.0.0.255 eq 13724
permit ip any 201.114.69.0 0.0.0.255
```

SOLUTION :

Reducing ACL size reduces ACL configuration errors

Configuration Errors are linearly proportional to ACL size

Optimization Approaches

Optimal ACL Definition

Optimal ACL is an ACL that meets all security requirements with the least amount of processing overhead.

- **Two main approaches to ACL optimization:**

1. **Reducing ACL Size:**
   1. Remove unnecessary rules.
      1. Remove Shadowed Rules Algorithm: $O(n^4)$
      2. Remove Covered Rules Algorithm: $O(n^4)$
   2. Combine a pair of rules into a single general rule
      1. Bit Combine Algorithm: $O(n^5)$

2. **Rearranging ACL rules:**
   1. Place rules with higher hits probabilities first without changing ACL semantics.
      1. Hits Optimizer Algorithm: $O(n^3 \lg n)$

Optimization Approaches

Optimal ACL Definition

Optimal ACL is an ACL that meets all security requirements with the least amount of processing overhead.

- **Optimizing ACLs is an NP-Complete problem [1].**
  - → **Heuristics solutions needed.**
  - → **Scenario-based optimization:**
    - ✓ **80% reduction in EPL**
    - ✓ **40% reduction in ACL size.**

(1) Source: V. Grout, J. McGinn, and J.Davies. "Real-Time Optimisation of Access Control Lists for Efficient Internet Packet Filtering", Journal of Heuristics, Vol. 12, 2006.

Quick Definitions

Rule Dependency:

$$Ri \,\Delta\, Rj \leftrightarrow (Rj.prtcl \cap Ri.prtcl \neq \phi) \wedge (Rj.sa \cap Ri.sa \neq \phi) \wedge (Rj.sp \cap Ri.sp \neq \phi) \wedge (Rj.da \cap Ri.da \neq \phi) \wedge (Rj.dp \cap Ri.dp \neq \phi)$$

Superset Rule:

$$Ri \supseteq Rj \leftrightarrow (Ri.prtcl \supseteq Rj.prtcl) \wedge (Ri.sa \supseteq Rj.sa) \wedge (Ri.sp \supseteq Rj.sp) \wedge (Ri.da \supseteq Rj.da) \wedge (Ri.dp \supseteq Rj.dp)$$

Shadowed Rule:

$$Ri \rhd Rj \leftrightarrow (i < j) \wedge Ri \supseteq Rj$$

Covered Rule:

$$Ri \lhd Rj \leftrightarrow (i < j) \wedge (Rj.act = Ri.act) \wedge Rj \supseteq Ri \wedge (\forall k \mid i < k < j : Not(Ri \Delta Rk) \vee (Ri.act = Rk.act))$$

Algorithmic Considerations

- Changing the location of rules and/or combining them may change ACL semantics.
  - ✓ Solution: Correct update of rules' boundaries after each change.
- Traffic counts are not always available for ACLs.
  - ✓ Solution: Use prediction factors.
- Some anomalies may be deliberate.
  - ✓ Solution: Report every action during optimization for the administrator to review.
- Data Fie

Action	Protocol	Src IP	Src WCM
Dst IP	Dst WCM	Src Port Rng	Dst Port Rng
Hit Counts + Prediction Factor	Rule Latency	Rule Up Bound	Rule Bottom Bound

General Algorithm

Go through the ACL & **Look for potential optimization**

Potential Optimization found?

→ **Within rules' boundaries** (i.e. no change in ACL semantics) **?**

→ **Perform change**

Report Action

Update Rules' boundaries

Update ACL general variables (e.g. pointers to first and last rules, ACL size …etc)

ارامكو السعودية
Saudi Aramco

First Approach: Reducing ACL Size

```
permit  ip      10.1.12.2  0.0.0.7     any
deny    ip      any                    any
```

- **Removing Shadowed Rules Procedure:**
  - ✓ **Rule 3 is shadowed by Rule 2 → Remove Rule 3**
- **Removing Covered Rules Procedure:**
  - ✓ **Rule 1 is covered by Rule 2 → Remove Rule 1**
- **Bit Combine Procedure:**
  - ✓ **Rule 1 and Rule 2 can be combined → Combine Rules 1 & 2**

ارامكو السعودية
Saudi Aramco

Second Approach: Hits Optimizer

```
permit  ip    10.1.12.2  0.0.0.3    any
permit  icmp  host 10.114.1.100     host 10.114.48.228    echo
permit  icmp  host 10.114.1.100     host 10.114.48.228    echo-reply
permit  tcp   any                   10.114.48.0 0.0.0.255  eq www (10020 matches)
deny    ip    any                   any (365144 matches)
```

- **Sorted based on each individual rule's Effective Hits Probability (EHP) and ACL semantics.**
- **EHP is a function of three variables:**
  1. **Individual Rule Latencies (RL)**
  2. **Hits Counts**
  3. **Prediction Factor (PF)**

$$EHP \propto (HitCounts + PF) / RL$$

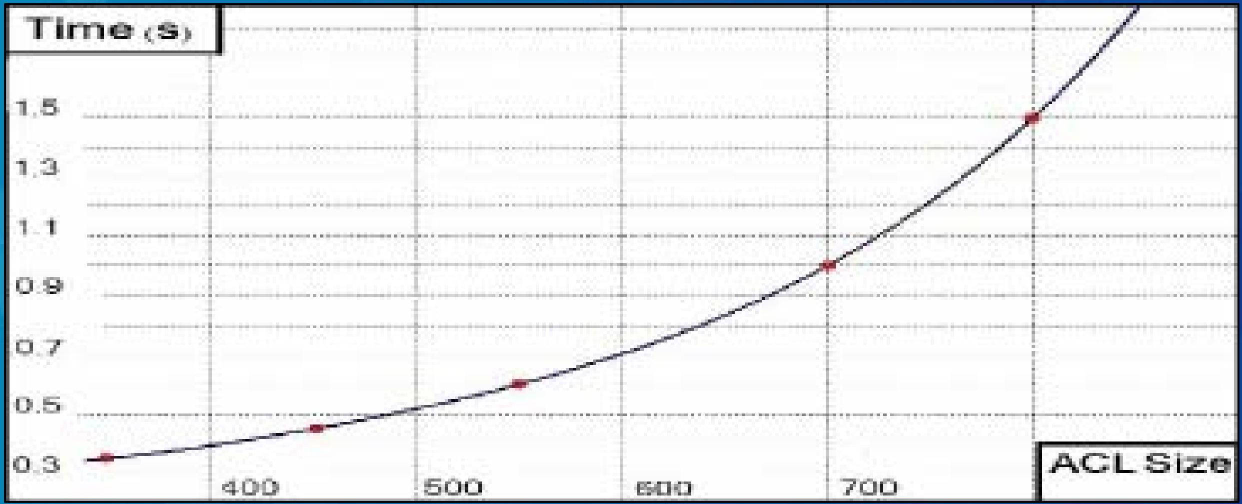
Second Approach: Hits Optimizer

```
1.  permit  tcp   any                    10.114.48.0 0.0.0.255  eq 80 (109 matches)
2.  permit  ip    10.1.12.2  0.0.0.3     any
3.  permit  icmp  host 10.114.1.100      host 10.114.48.228         echo
4.  permit  icmp  host 10.114.1.100      host 10.114.48.228         echo-reply
5.  deny    ip    any                    any (365144 matches)
```

- **Assuming Rule Latencies are equal:**
  - ✓ **Rule 4 is placed first, highest HitsCount.**
  - ✓ **Rule 5 cannot be relocated to keep ACL semantics.**
  - ✓ **Rules 1, 2, & 3 are sorted based on each rule's prediction factor**
    - **Rule 1 has a higher prediction factor than rules 2 and 3 for two reasons:**
      - i. **IP protocol vs. ICMP**
      - ii. **"destination = any" vs. "destination = host"**

ACLO Application: Quick Facts

- **Command-line C++ application.**
- **3000+ lines of source code.**
- **200KB executable file.**
- **Time complexity is $O(n^5)$**
  - **Customizable for speed at the expense of efficiency.** E.g.
    - **Selected scenarios instead of all scenarios.**
    - **Controlled number of iterations for loops instead of**



ارامكو السعودية
Saudi Aramco

ACLO Optimization Results

- **100+ ACLs optimized:**
  - **Average of 80% reduction in EPL (i.e. 5-time improvement)**
  - **Average of 40% reduction in ACL size (i.e.. ≈ 2-time improvement)**
  - **dure and**

Reduction	Shadowed Rules Removal	Covered Rules Removal	Rules Combining Procedure	Hits Optimizer
EPL	1%	10%	25%	77%
Size	2.5%	5%	37%	0%

Reasons:

i. **Shadowing and covering are easier to detect by network administrators, and normally considered during security specifications.**

ii. **Bit Combining procedure involves an excellent**

ارامكو السعودية
Saudi Aramco

ACLO Application Demo

Conclusions

- Optimizing ACLs is necessary to reduce processing overhead and configuration errors.
- It can be easily implemented using heuristic fast and efficient algorithms that reduces the size of the ACL and/or rearranges its rules.
- Hits Optimizer and Bit Combine procedures yield the greatest bulk of optimization.
- Algorithms can be easily customized in terms of time vs. efficiency, and can be implemented partially or fully, both online and offline.

Q & A