# Improving Attack Graph Visualization through Data Reduction and Attack Grouping

John Homer[1], Ashok Varikuti[1], Xinming Ou[1], and Miles McQueen[2]

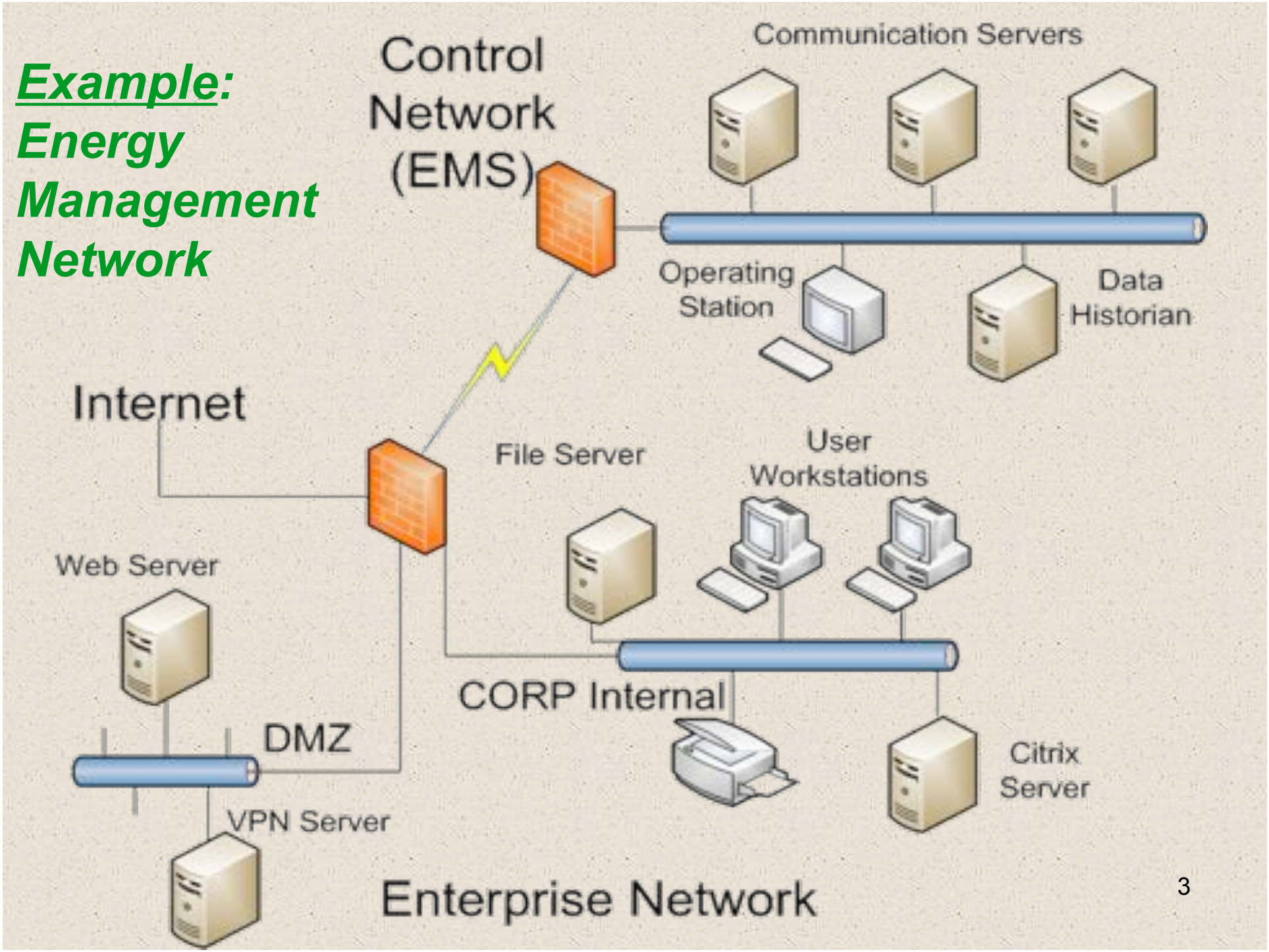
[1] Kansas State University

[2] Idaho National Laboratory

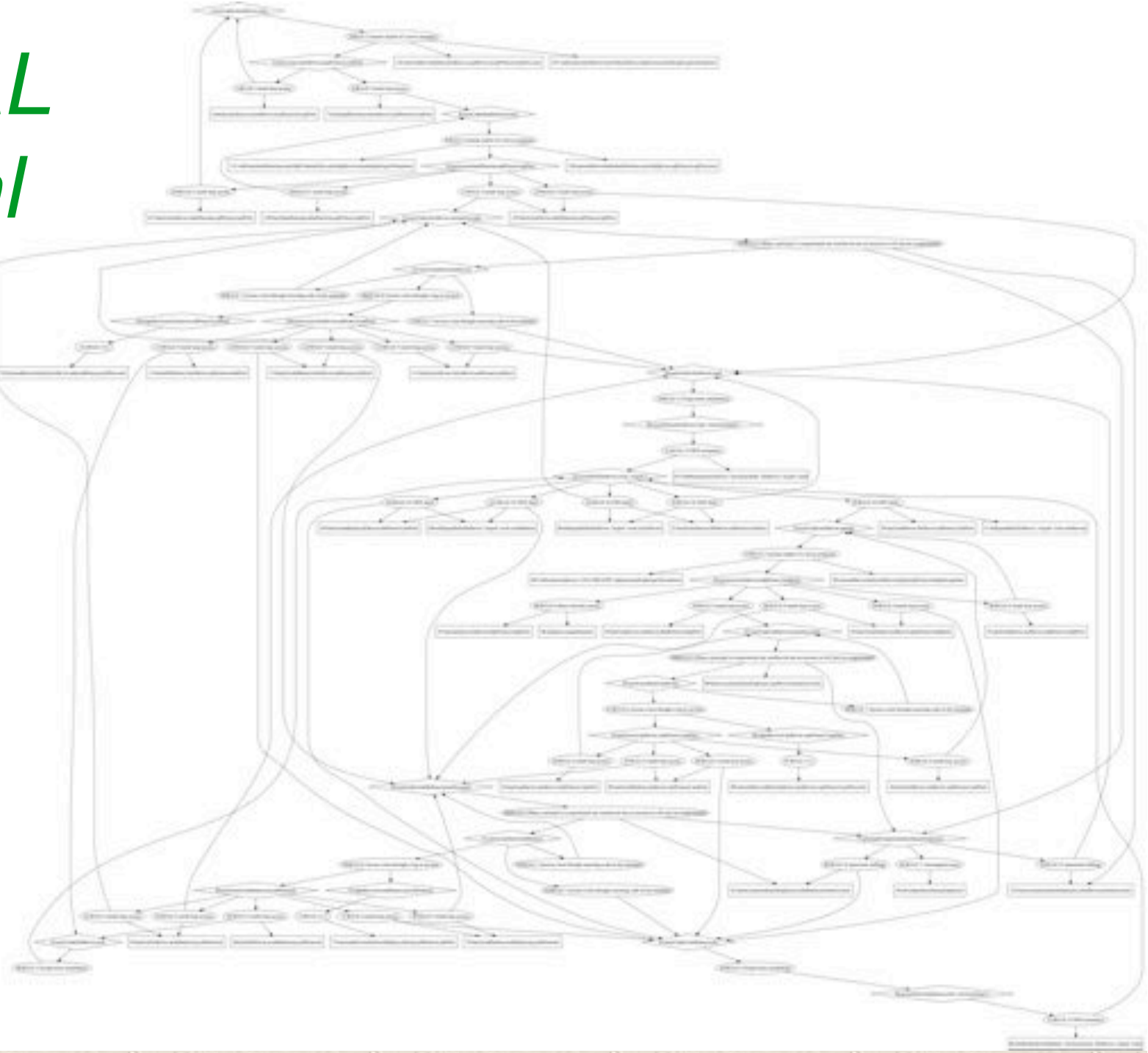
# Contributions

- Attack Graph Problems: Size & Complexity
  - Difficult to quickly identify most important data
  - Difficult to assess and act on complete set of possible attack steps
- Solutions :
  - Eliminate “useless” attack steps in graph
  - Add abstract nodes, representing exploits, to enable simpler identification of issues

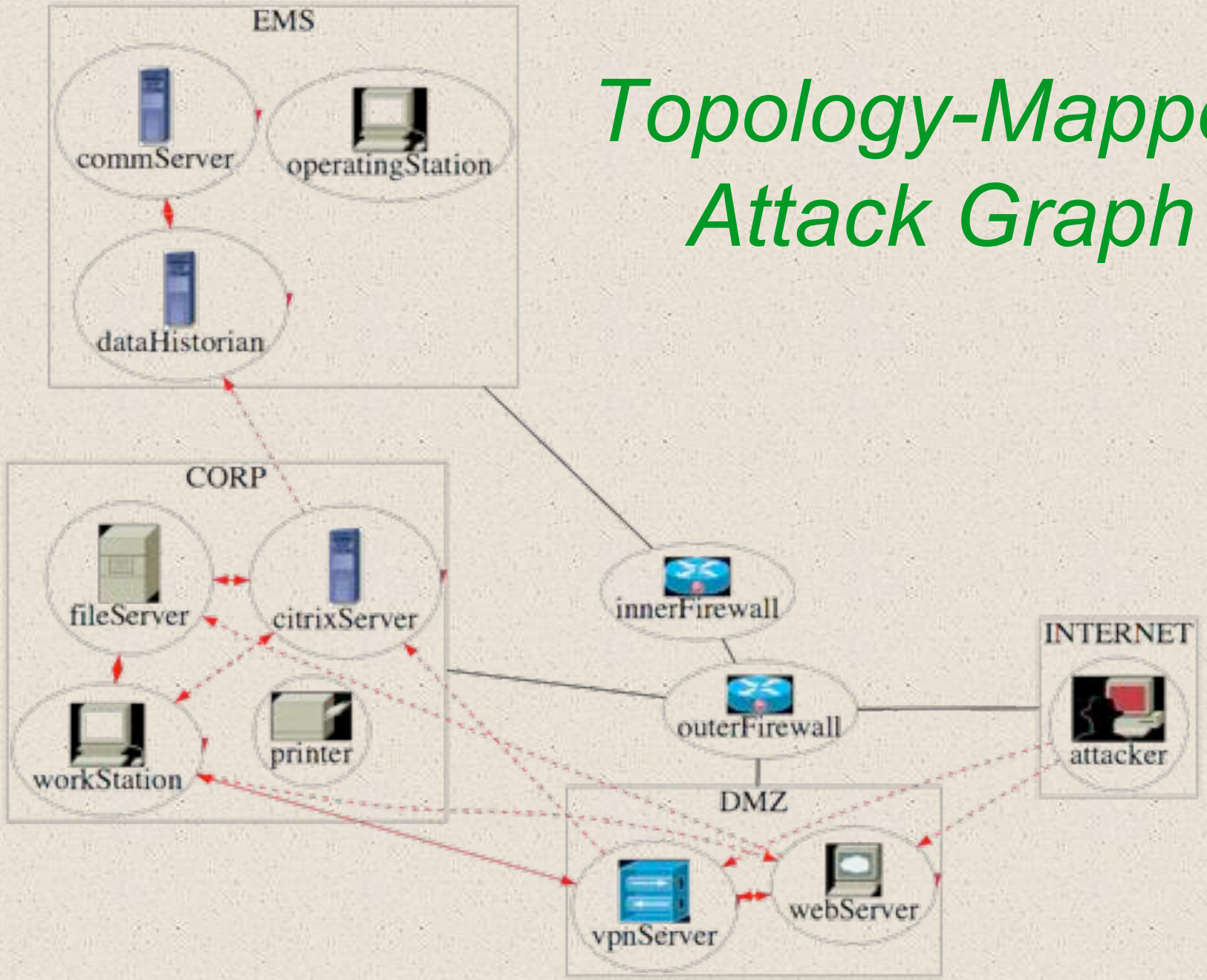
***Example:***
***Energy Management Network***



# MulVAL Logical Attack Graph



# Topology-Mapped Attack Graph



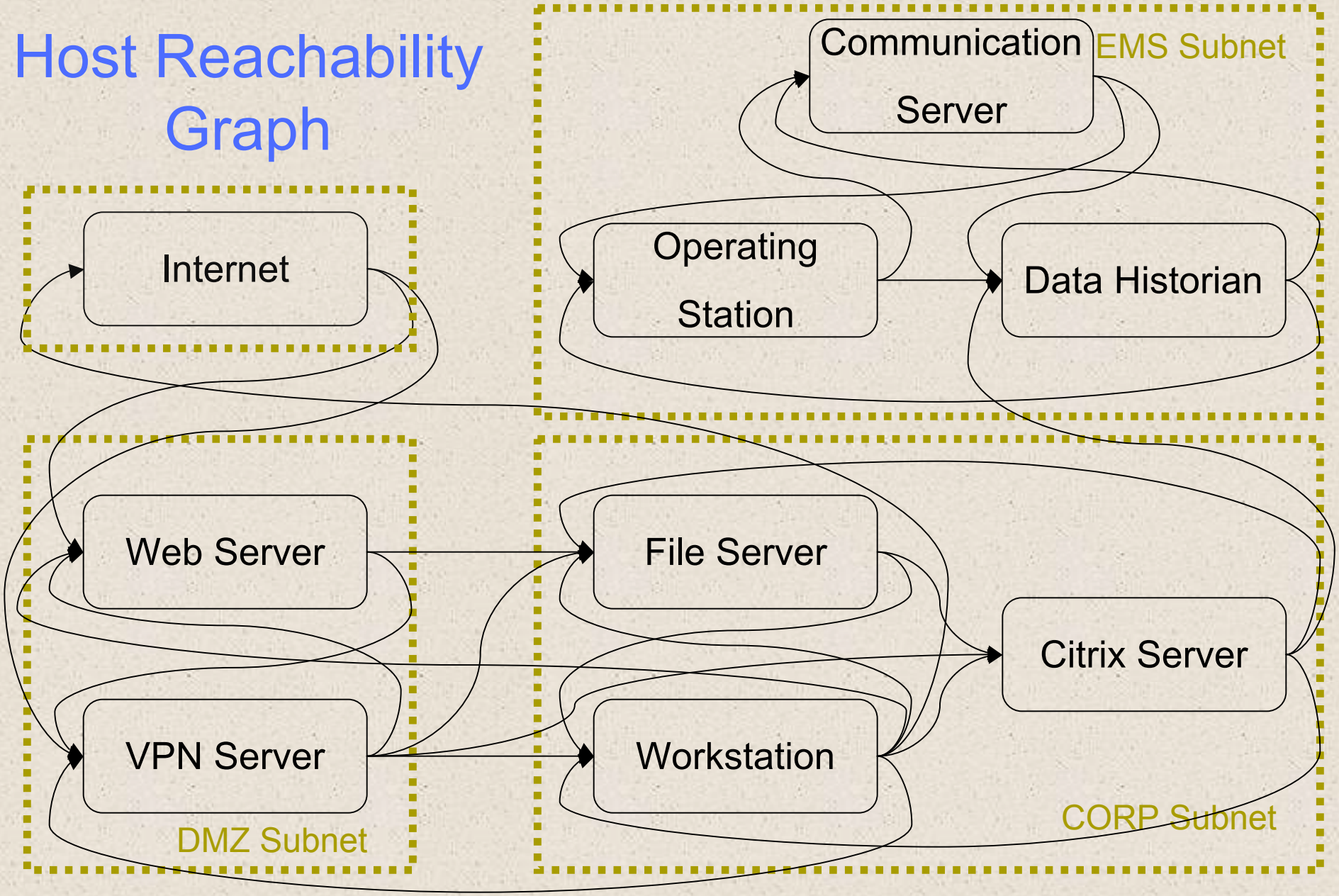
# “Useless” attack steps

Not all valid attack steps are useful

in quickly evaluating overall security

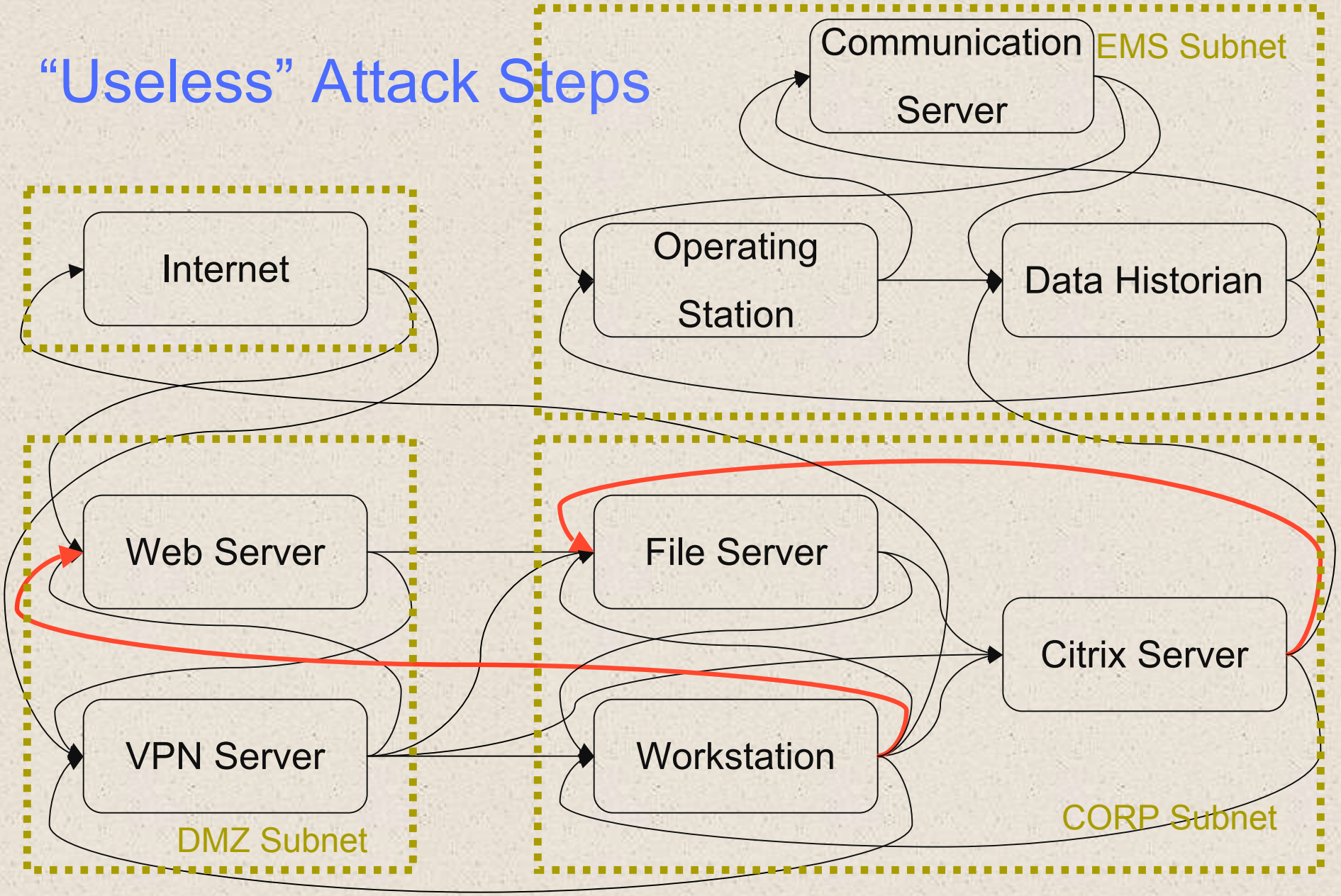
An attack step that does not enable a straightforward path to the goal privilege will be considered “*useless*”

# Host Reachability Graph




Are all of these transitions “useful”?

# “Useless” Attack Steps



# Trimming Algorithm

- Consider network topology at two levels:
  - "High-level" view of subnets within network
  - "Low-level" view of individual host machines within each subnet
- Trim edges differently at each level
  - Inter-subnet edges
  - Intra-subnet edges

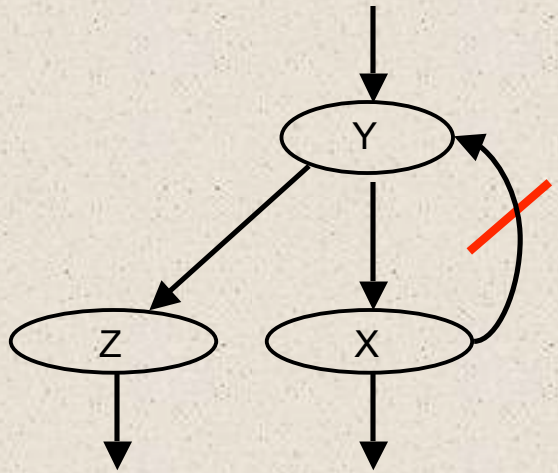
# Trimming Algorithm

- Trimming on inter-subnet edges
  - Trim “useless” edges based on dominator tree derived from graph of inter-subnet connectivity
- Trimming on intra-subnet edges
  - Trim “useless” edges based on potential expansion of attacker access to other subnets

# “Uselessness” by Domination

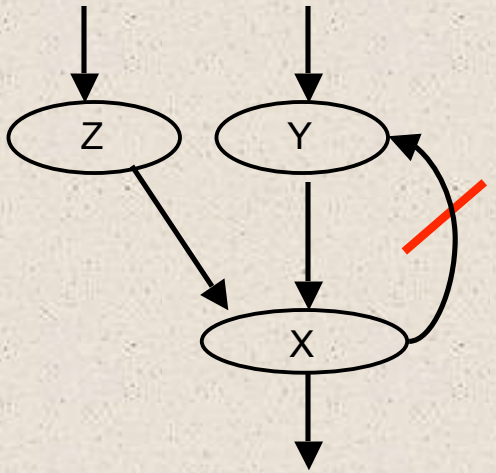
Trim edges X - Y when:

Y dominates X



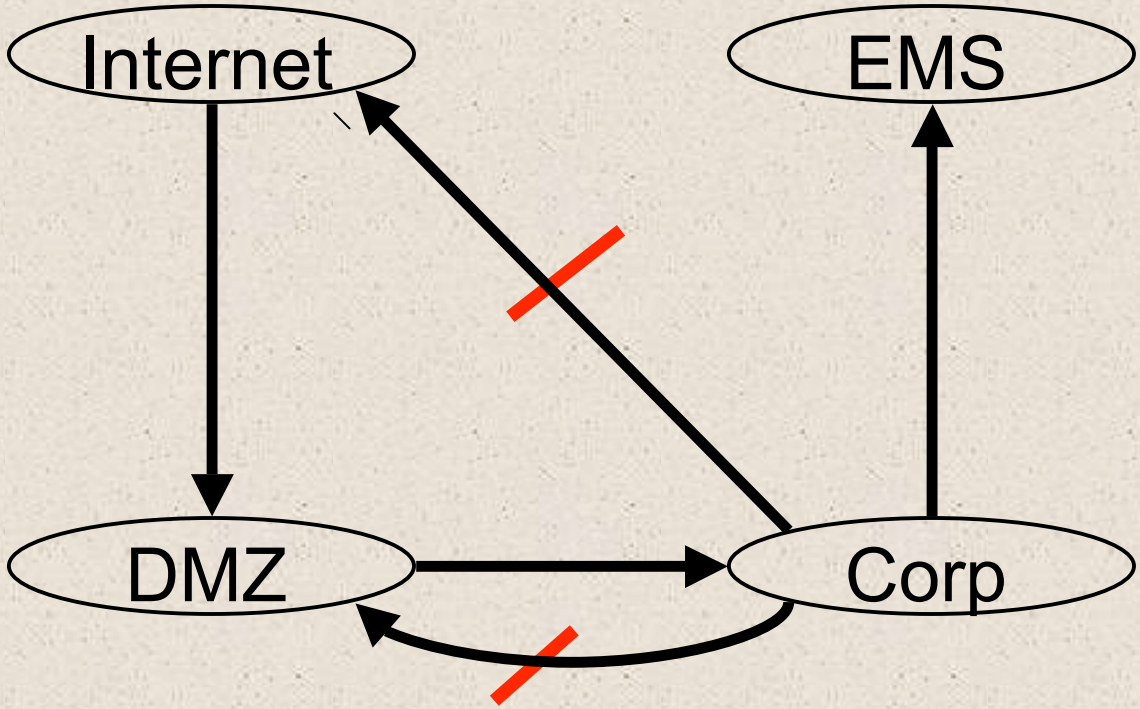

An edge from X to Y is useless because Y must have been already visited

X post-dominates Y



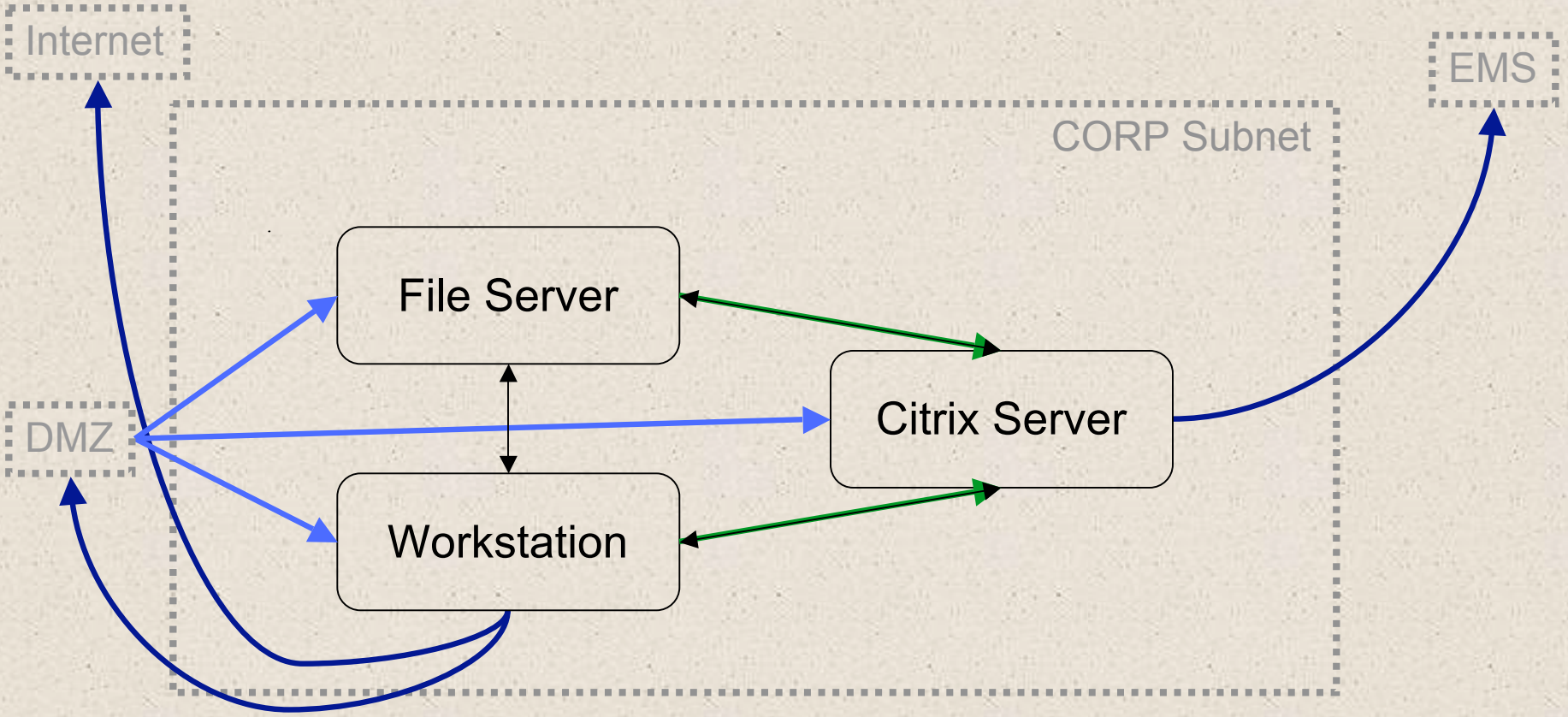

An edge from X to Y is useless because any path from Y must return to X

# Inter-Subnet Edge Trimming



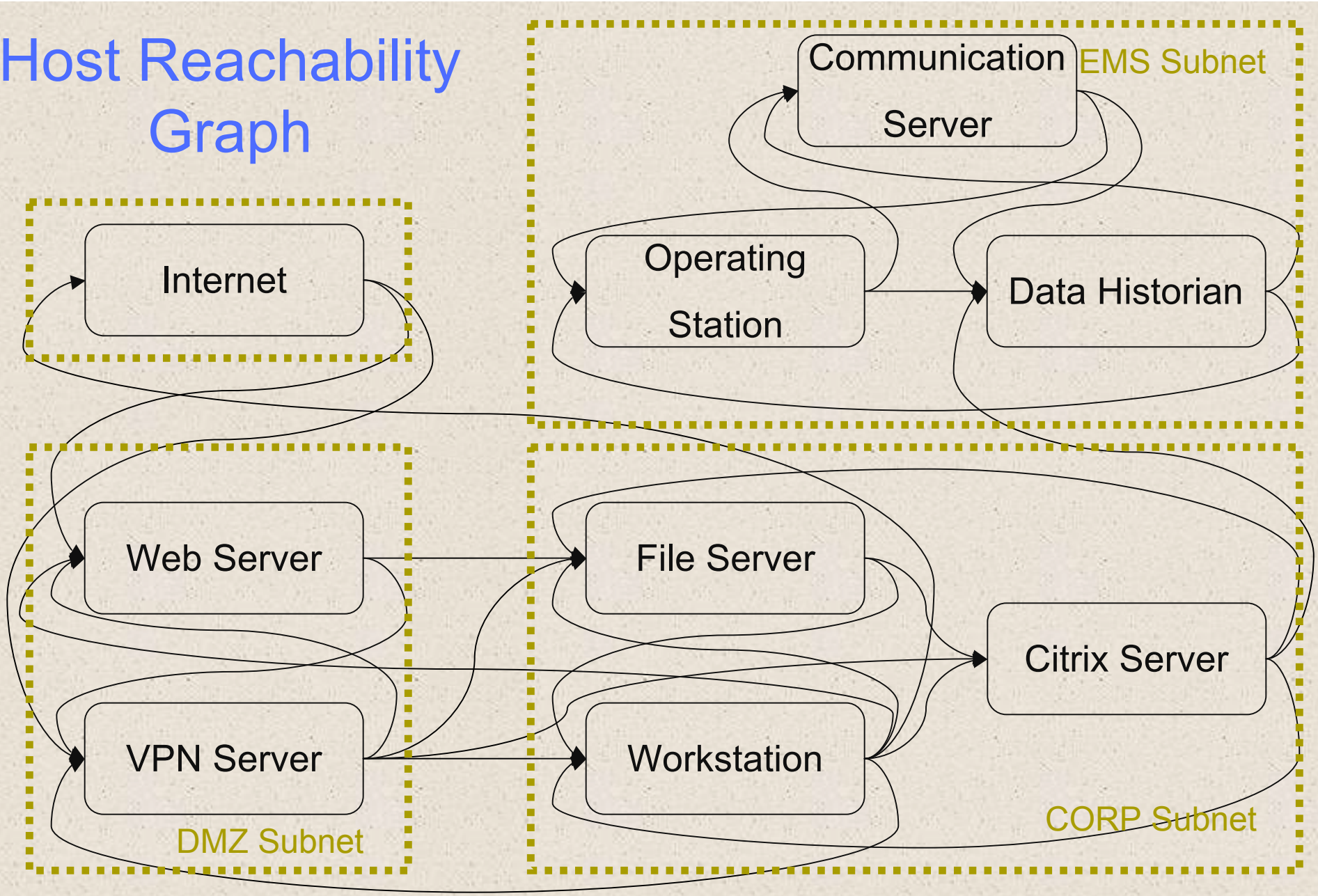

- Group host machines into distinct subnets
- Use host-access lists to determine inter-subnet transitions
- Block transitions X to Y when:
  - Y dominates X
  - X post-dominates Y

# Intra-Subnet Trimming

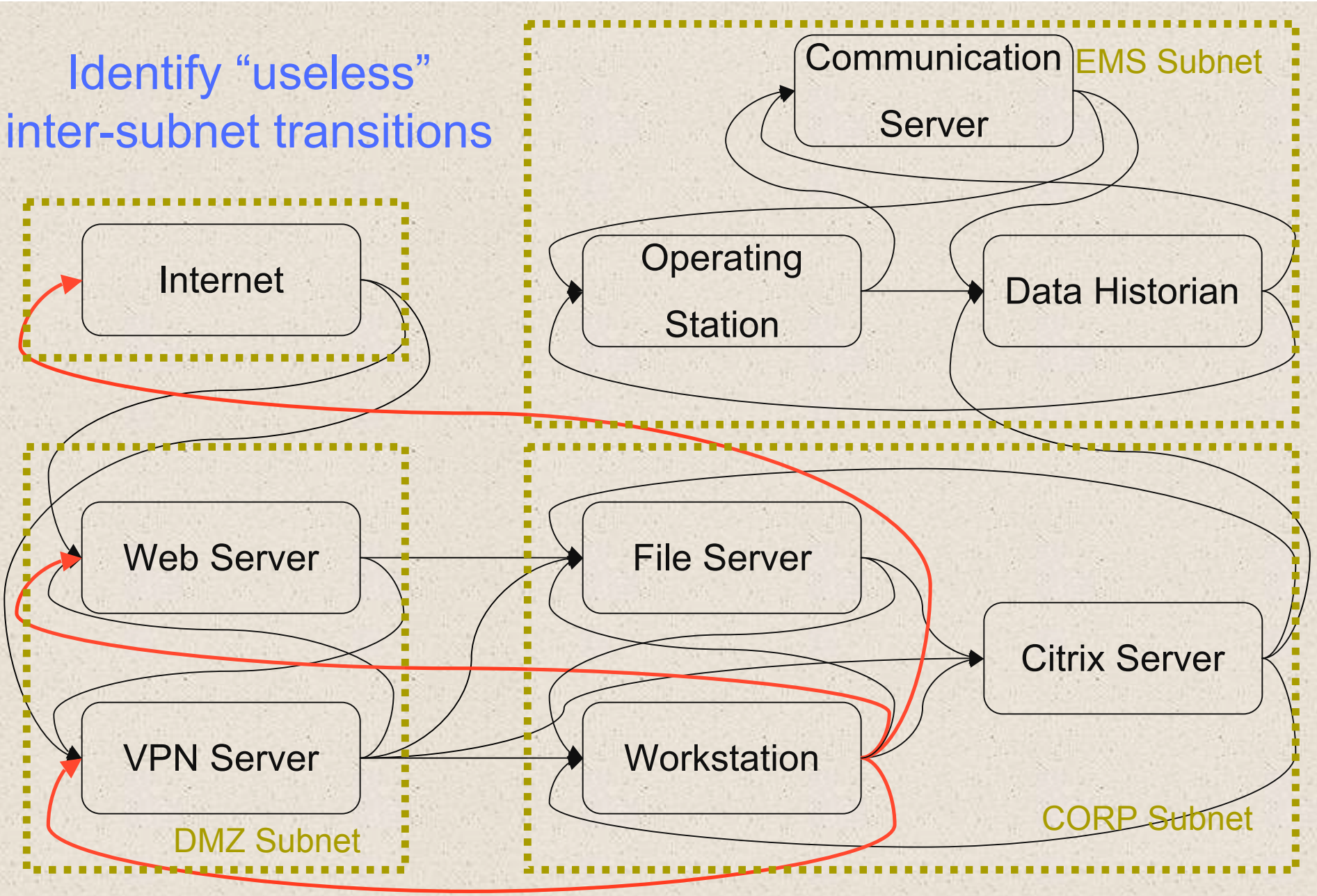



Allow intra-subnet transitions H to J only where J is a goal node, or where J has outgoing access to a “useful” subnet not directly accessible from H

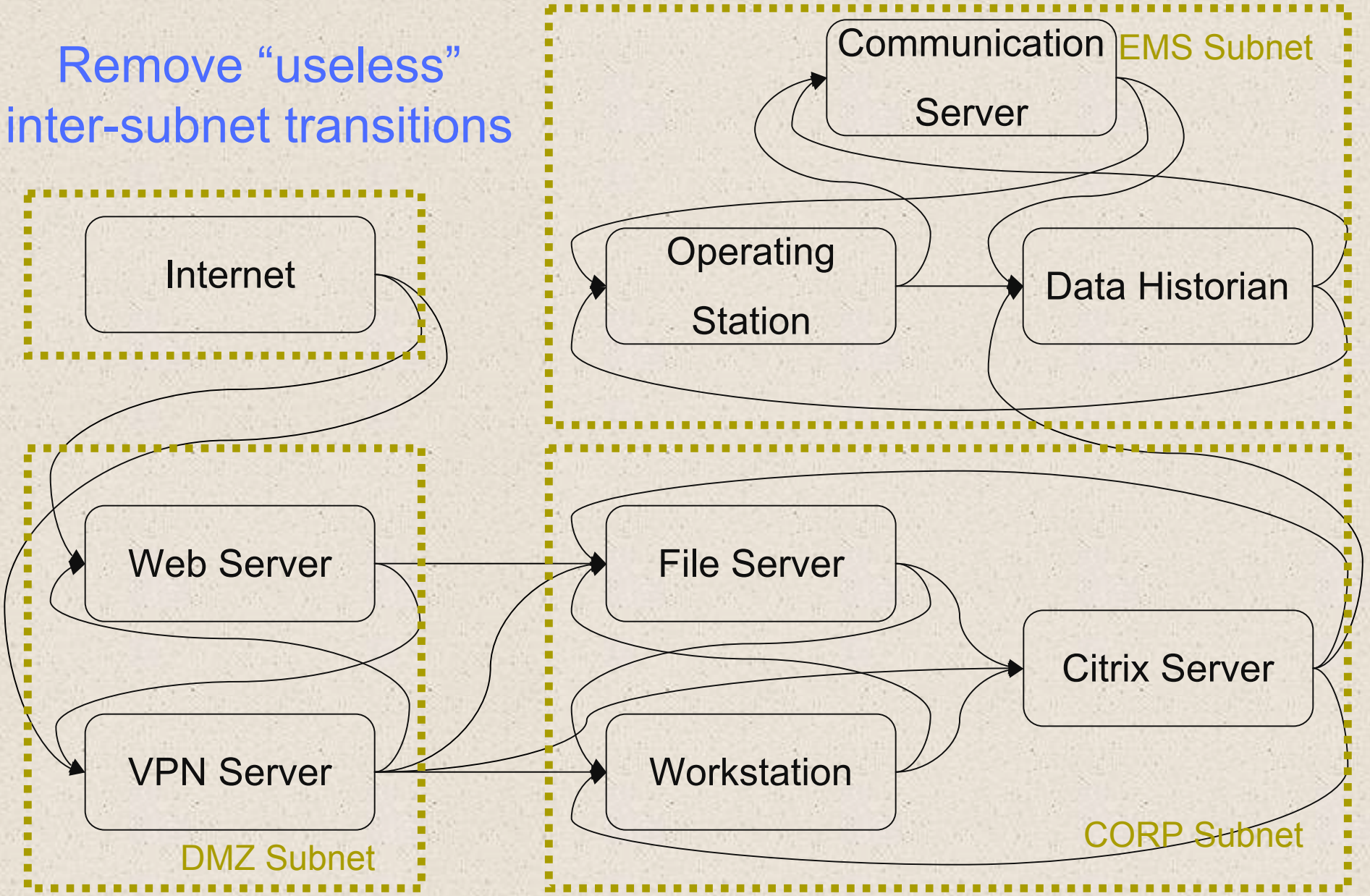
# Host Reachability Graph



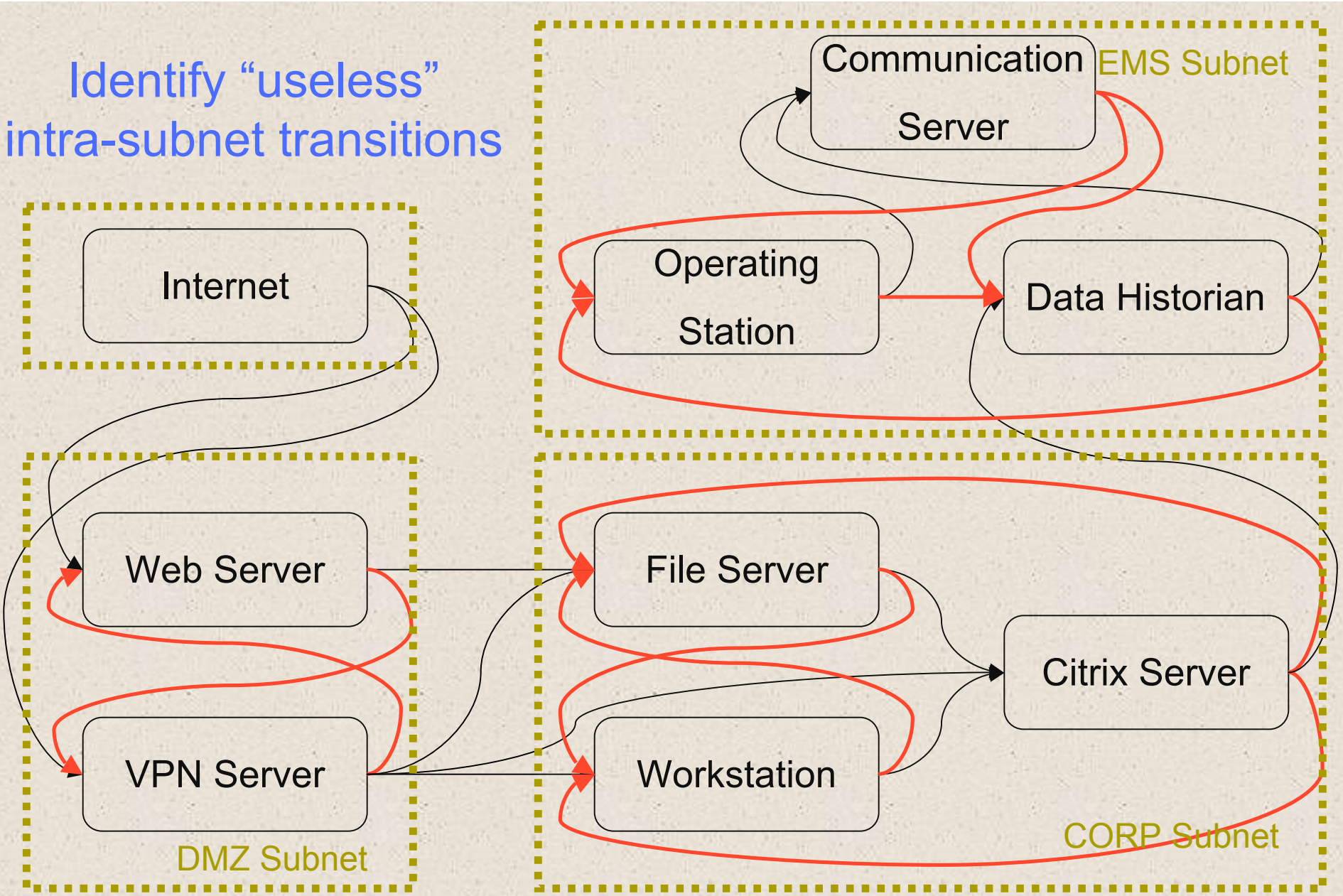
# Identify “useless” inter-subnet transitions



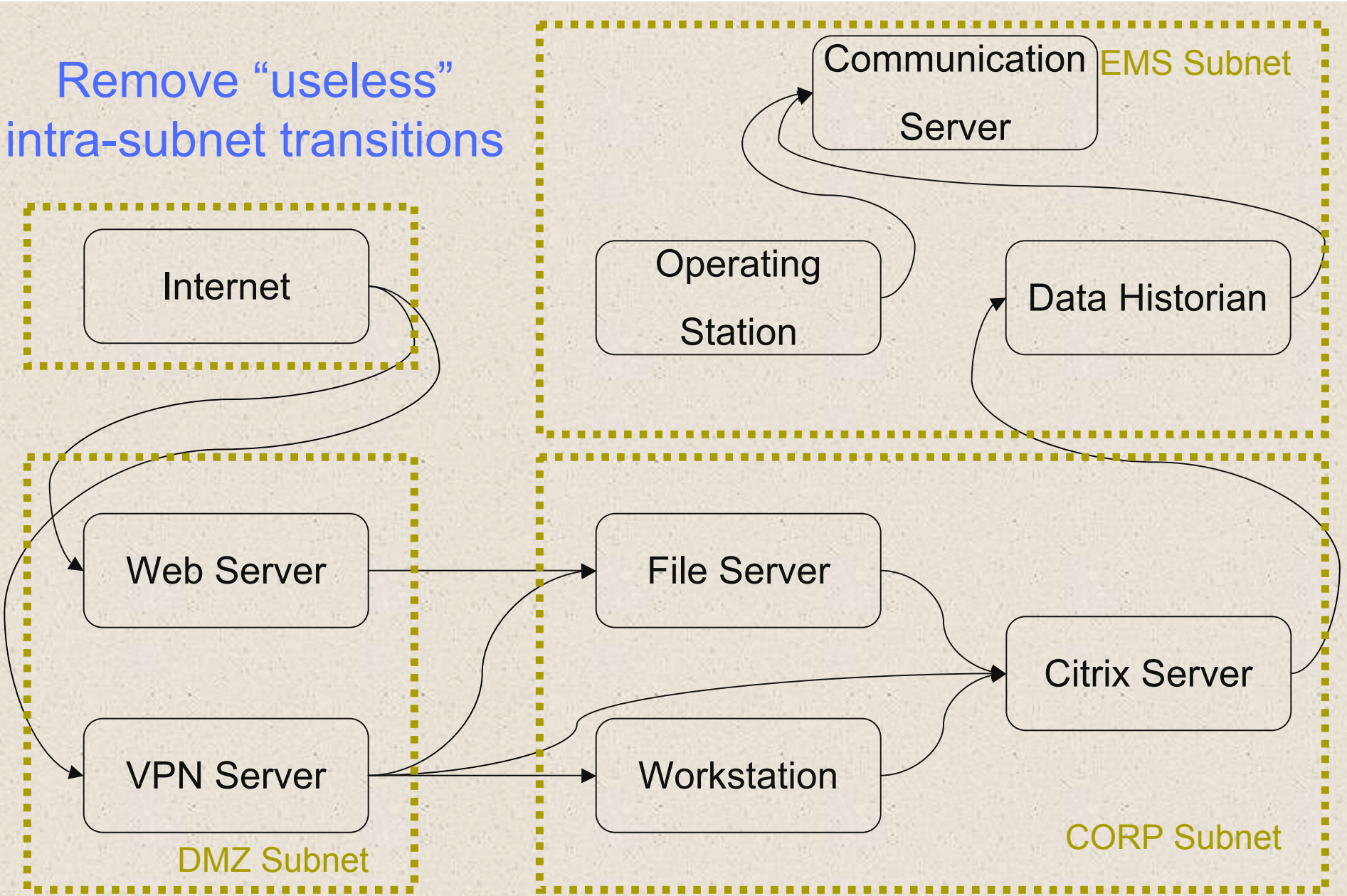
# Remove “useless” inter-subnet transitions



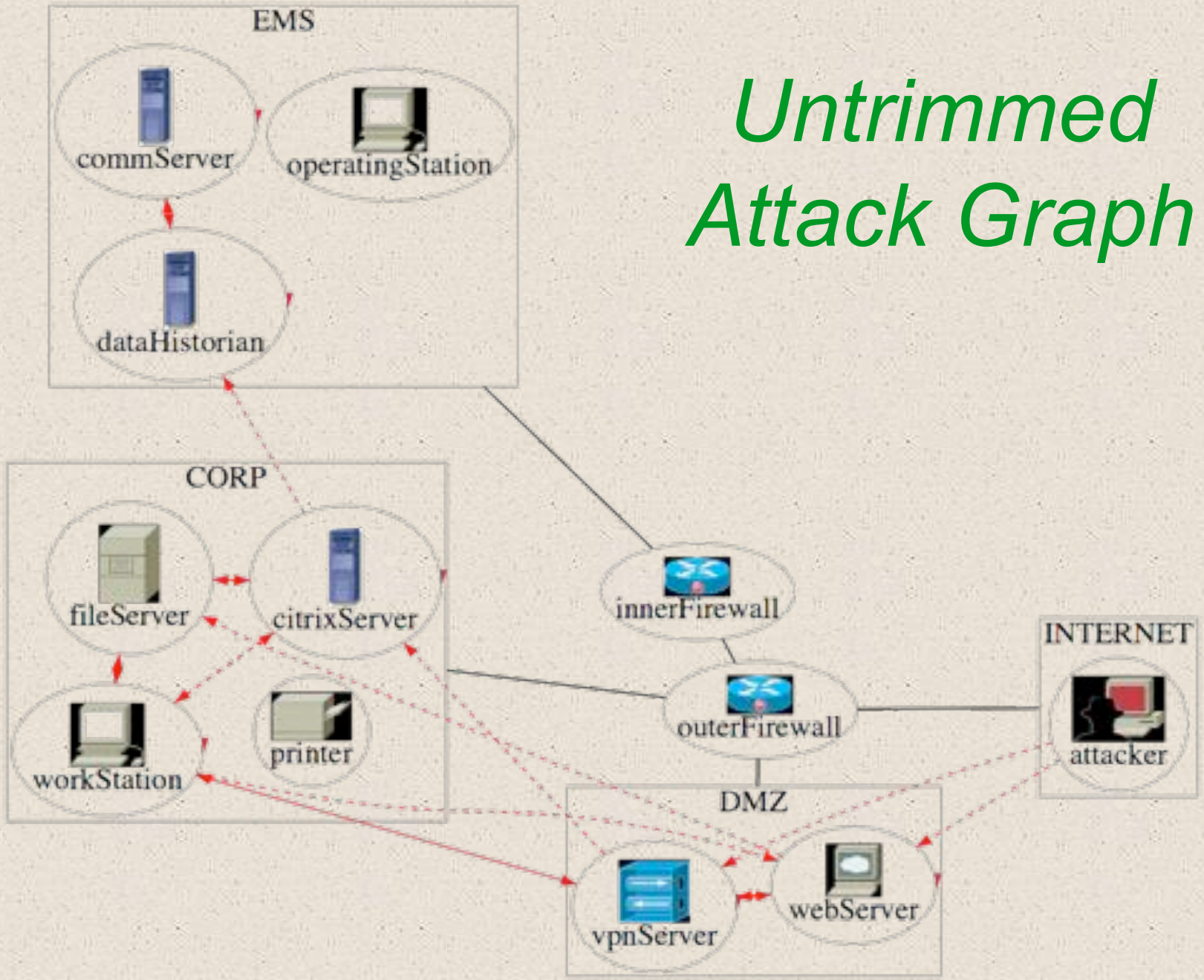
# Identify “useless” intra-subnet transitions



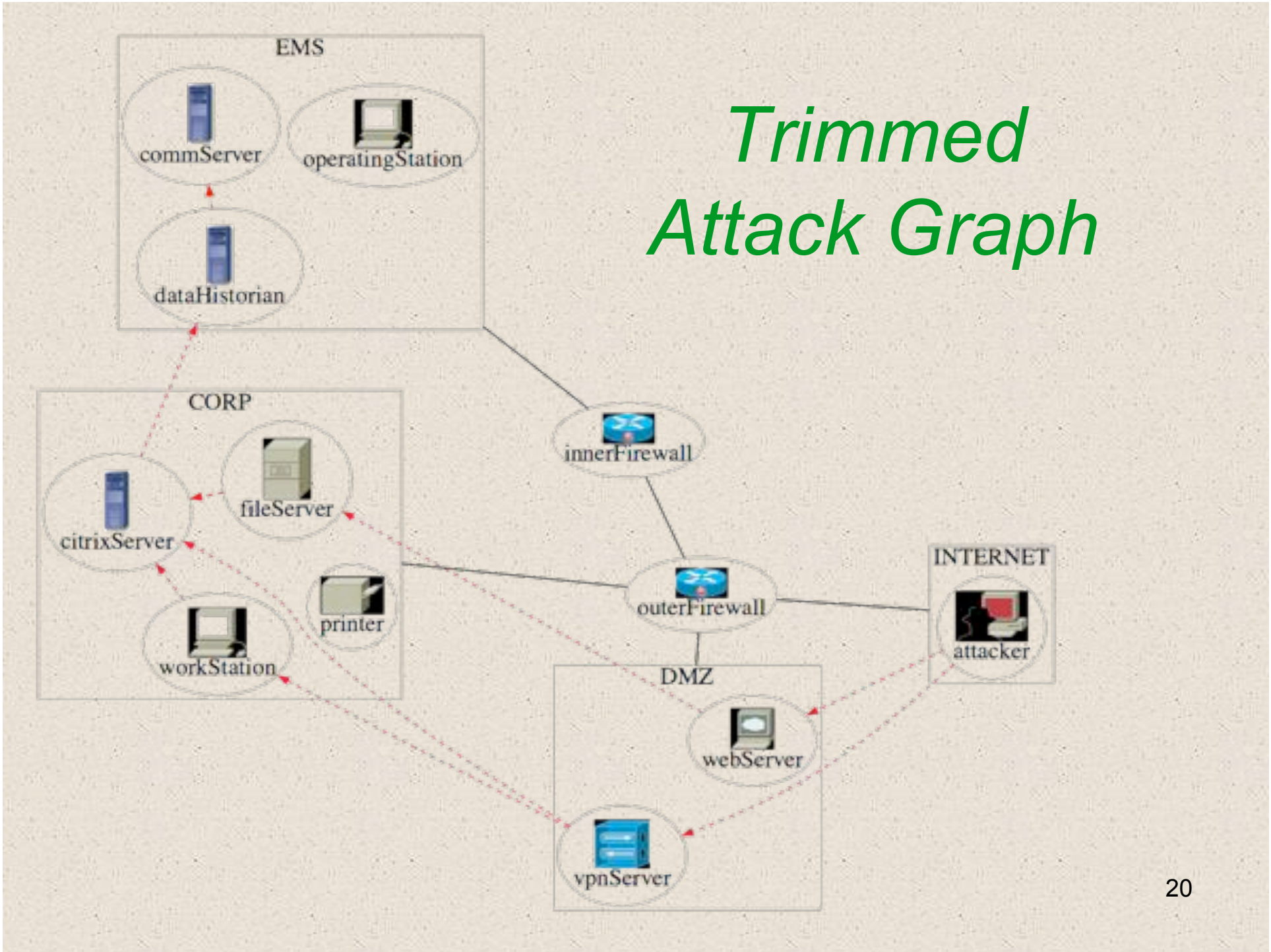
# Remove “useless” intra-subnet transitions



# *Untrimmed Attack Graph*



# Trimmed Attack Graph



# Benefits of Trimming

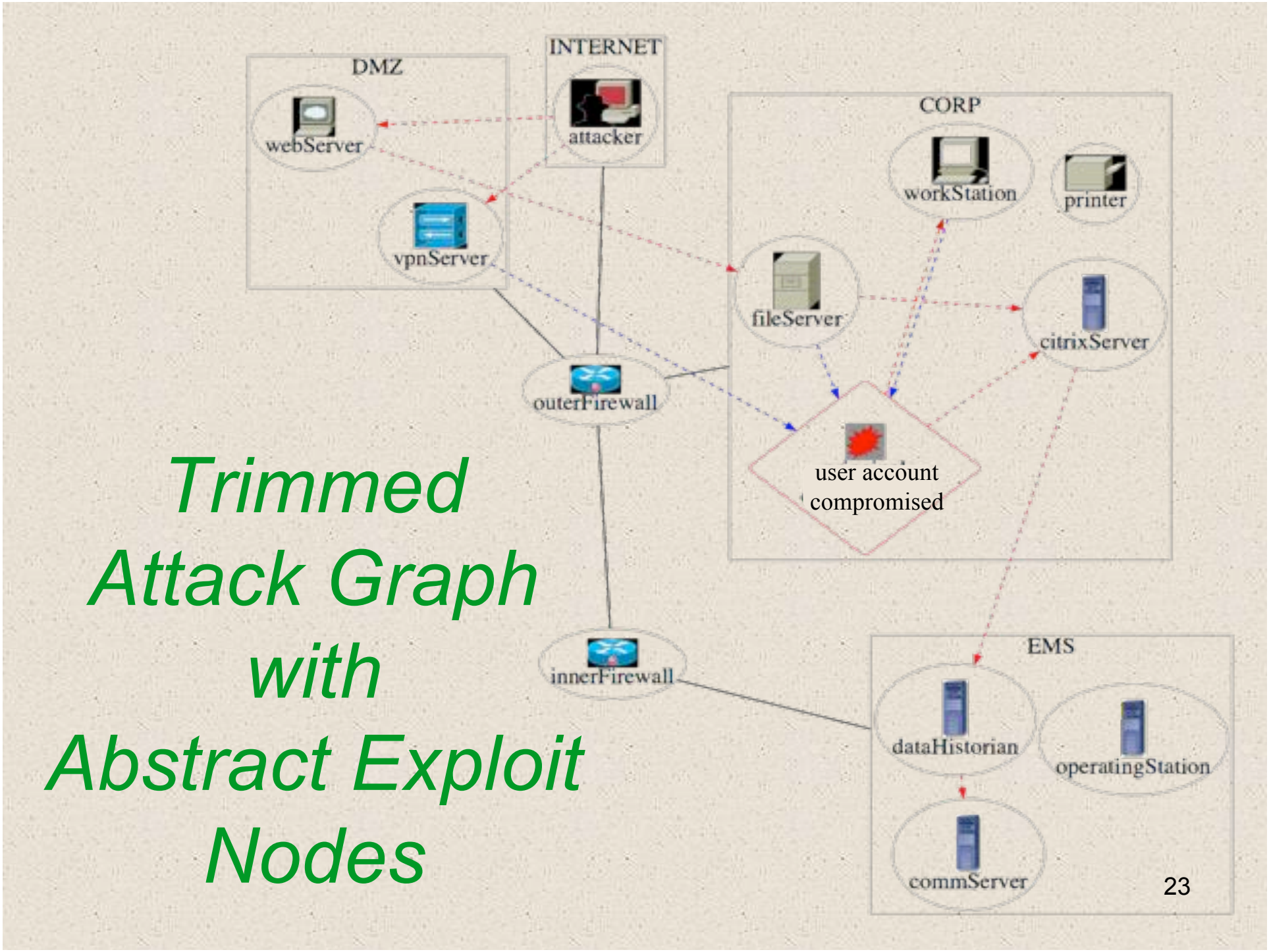
- Reduced data in attack graph
- Increased toolkit scalability
- Retained all “useful” attack paths
  - Internet - webServer - fileServer - citrixServer
  - Internet - vpnServer - workStation - citrixServer
  - Internet - vpnServer - citrixServer

    ( and then only one path from citrixServer )

# Exploit Abstraction

A simple topology mapping, even trimmed, can still hide full effect of each exploit

To counter this, we create a virtual node in the topology graph for each multi-source/multi-destination exploit




# *Trimmed Attack Graph with Abstract Exploit Nodes*

# Summary

Together, these improvements -
data reduction and exploit abstraction -
can increase the
accessibility and usability of the
data within an attack graph