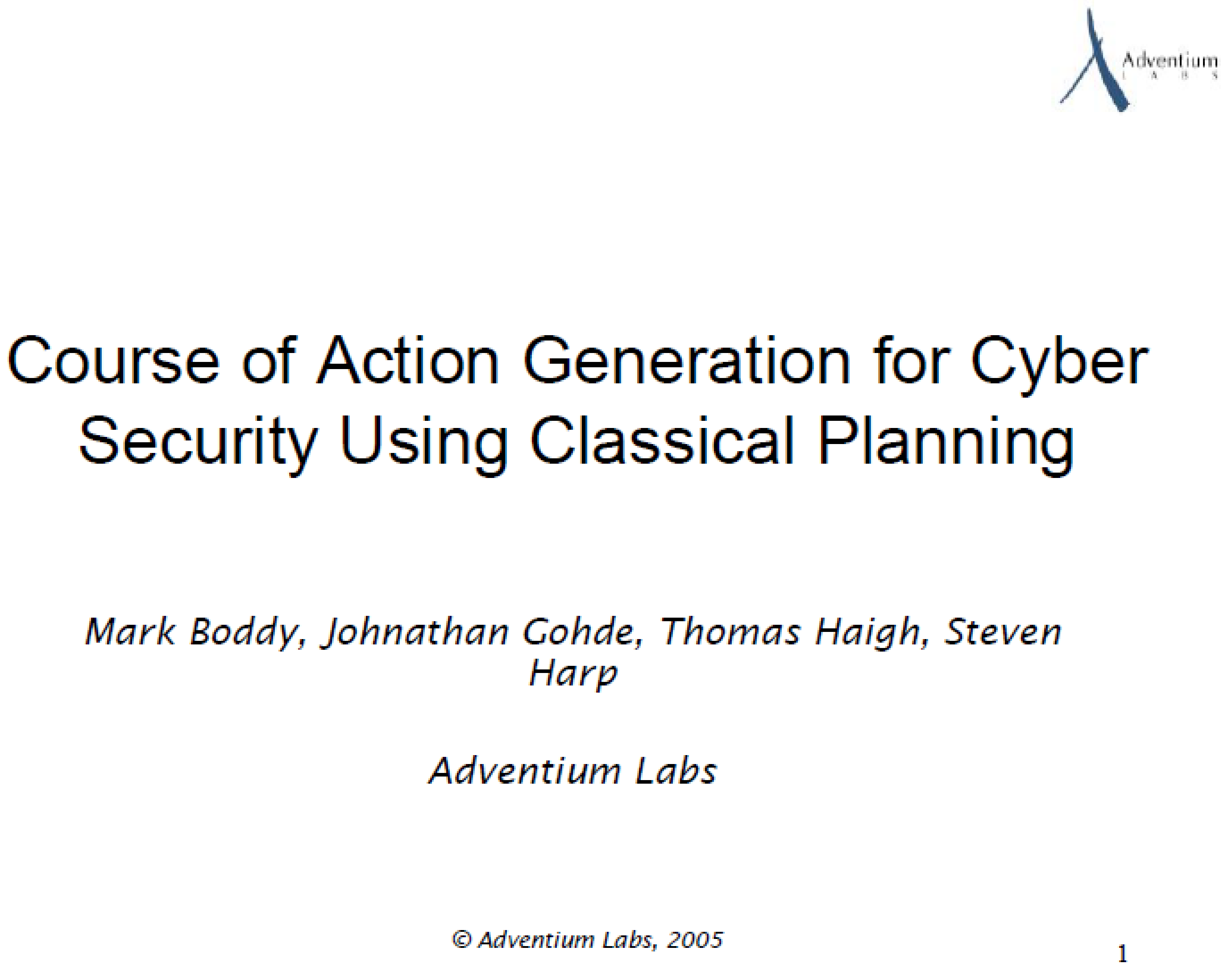


Course of Action Generation for Cyber Security Using Classical Planning

Mark Boddy, Johnathan Gohde, Thomas Haigh, Steven Harp

Adventium Labs

Presented by: Pingal Sapkota

The Problem

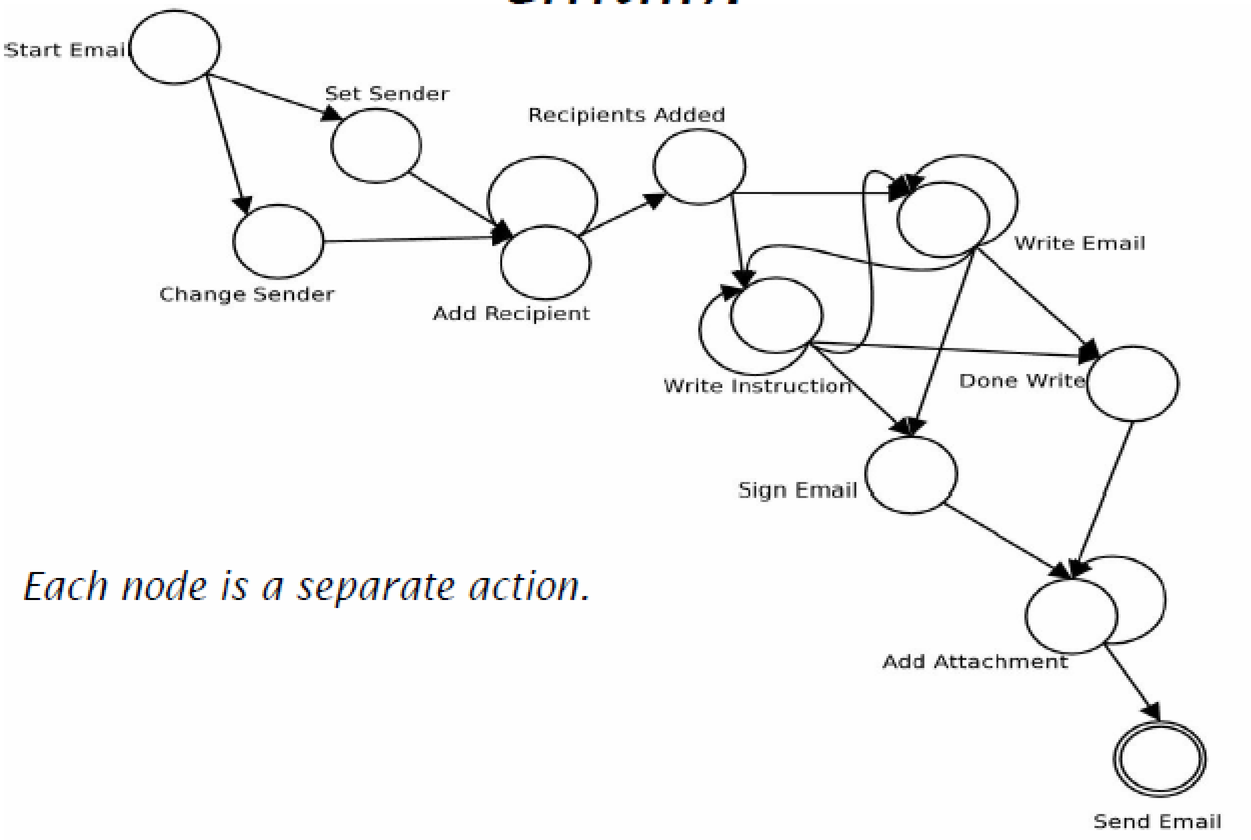
Finding and closing (or monitoring) attack vulnerabilities

For example:

1. Attacker sends an email message, spoofed to be from a colleague, with a new screensaver as an attachment.
2. Attachment is an executable that enables remote login, and captures and relays the users password.
3. Attacker logs into the machine and executes a buffer overflow attack, gaining root (admin) privileges.

4

Representing processes (e.g., composing and sending email).

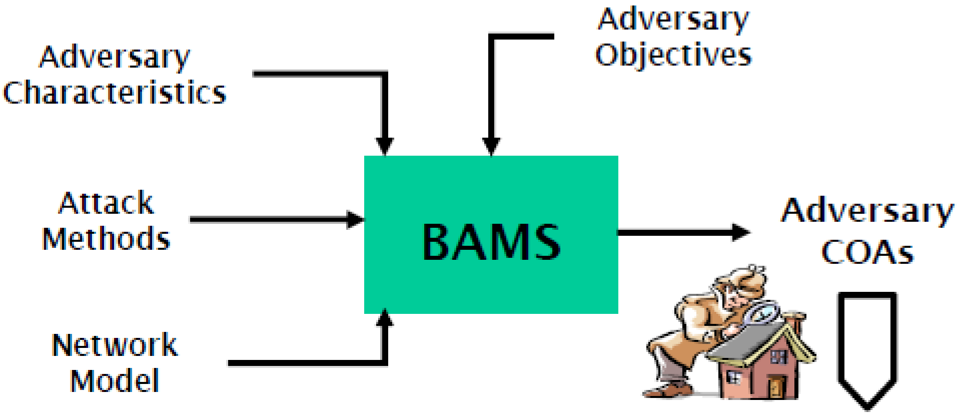


Each node is a separate action.

Why is this hard?

- Network and system scale, complexity, and dynamism
- Attackers are stealthy
- Many steps in any given attack may be legitimate.
- Some exploits involve actions taken outside the network.
- Some exploits are impossible or expensive to detect.
- Limited supply of experts

Behavioral Adversary Modeling System


Adversary Characteristics

Adversary Objectives

Attack Methods

BAMS

Adversary COAs

Network Model

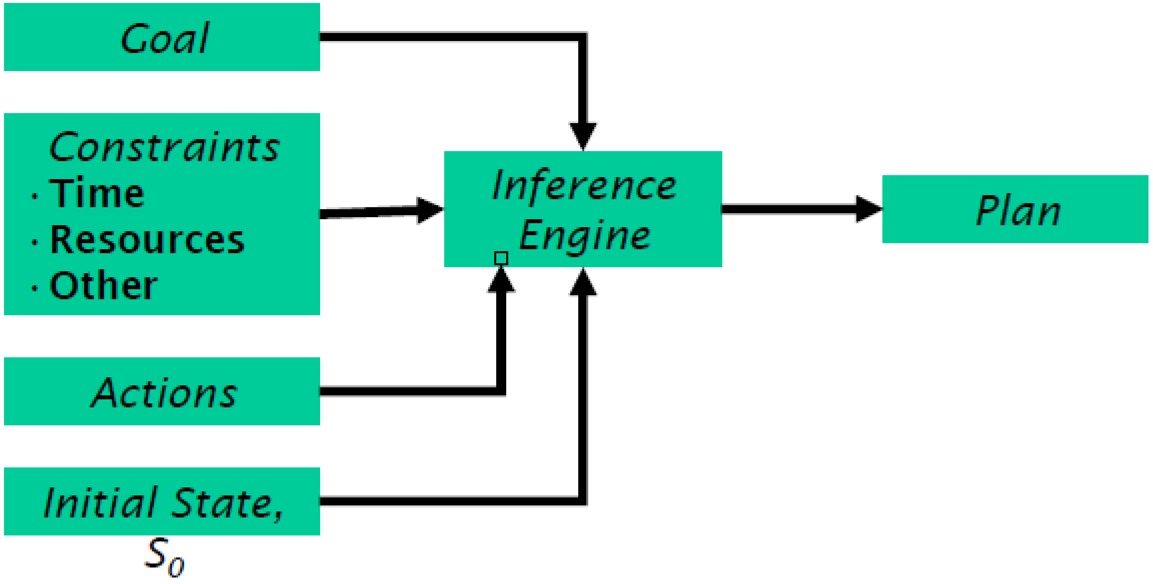
- Probability of success
- Probability of detection
- Likely side-effects
- Choke-points for prevention and detection
- Possible coalitions

Proposed solution: use classical planning

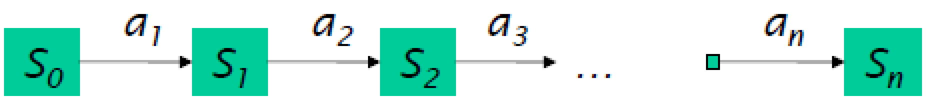
Scale of a Typical BAMS Problem

PROBLEM:	`NESACL’
Defined classes:	28
Defined predicates:	123
Number of objects:	100
Number of facts:	189
Number of goals:	1
Number of actions:	56

Classical Planning



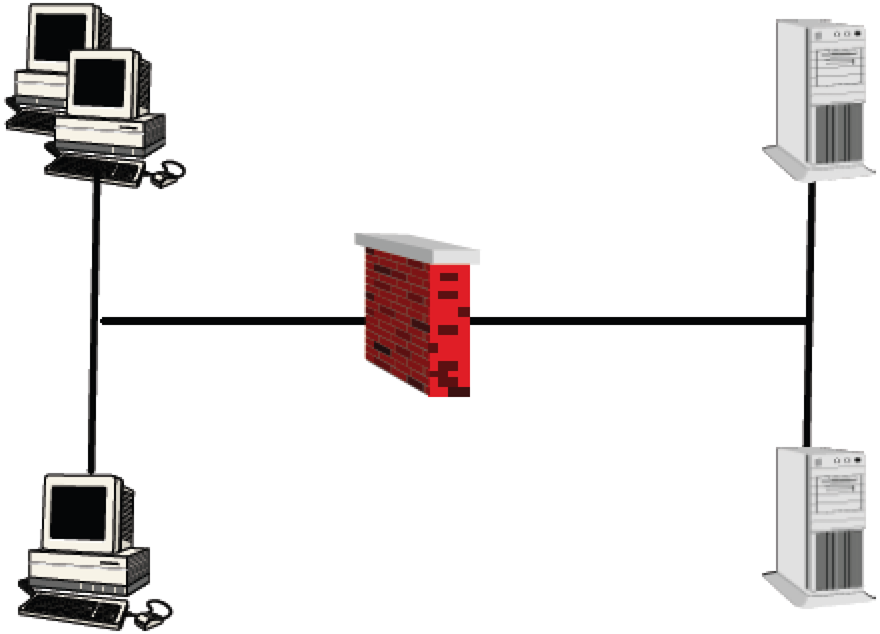
Plan:



Goal (S_n) = True

End-users

Mail Server



Sys Admin

- Password protected account
- Manages user accesses

COI Web Server

- SSL with fixed passwords
- ACLs

Domain Features

- *Cyber defenses:*
  authentication (2 forms), access permissions, controlled change of access permissions, firewalls, detectability, hubs and switches
- *Cyber exploits:*
  manipulation of access permissions, direct attacks against a workstation, password hacks, mis-directed trust (multiple aspects), host and network sniffing, spoofing, e-mail viruses, misdirected information,
- *Physical system and exploits:*
  location, shoulder surfing, hardware keystroke logger
- *Social behavior:*
  various forms of trust, social engineering, tolerance for risk, coalitions of attackers

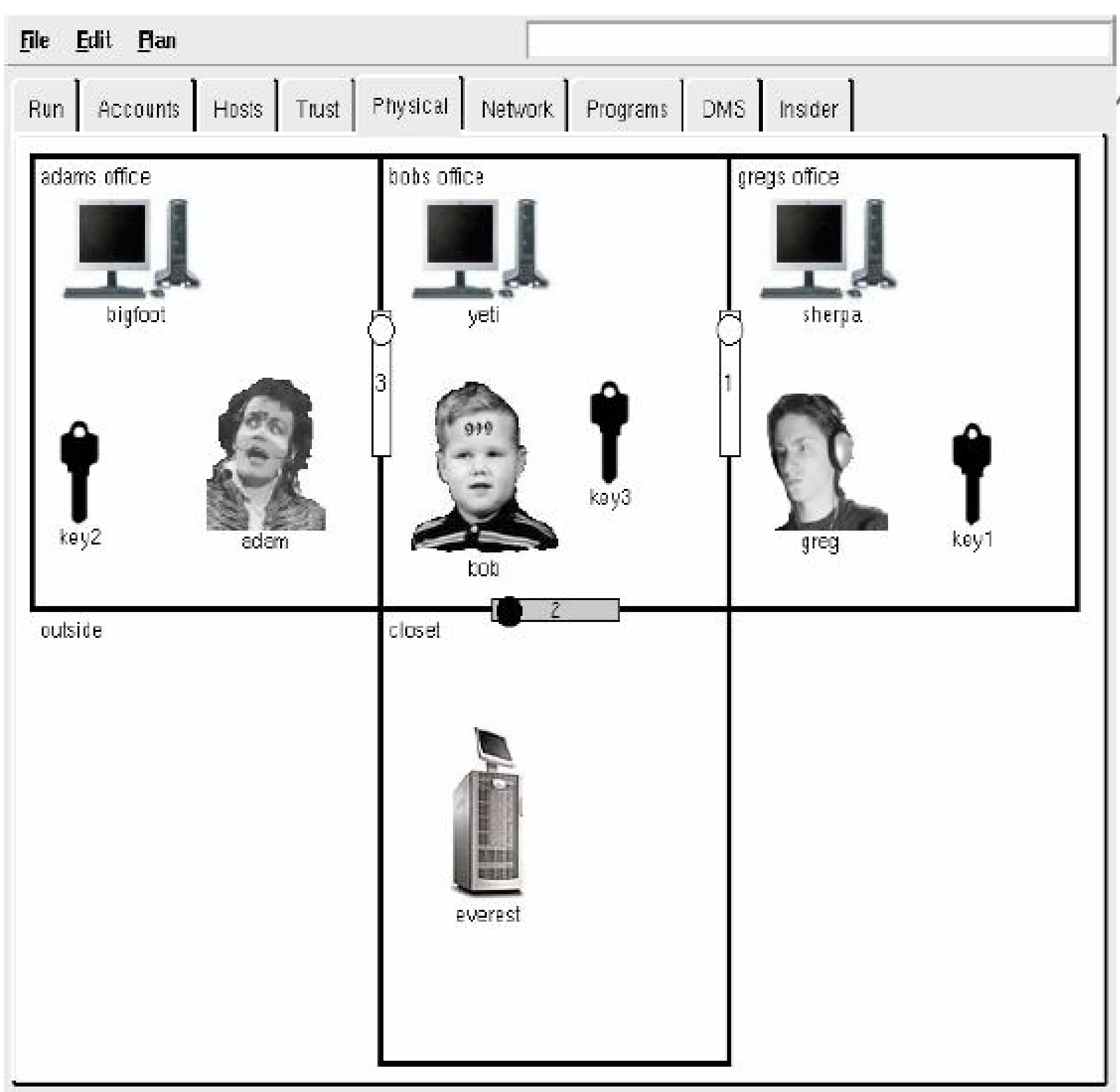
Examples: Facts

- *(insider bob)*
- *(in_room bob bobs_office)*
- *(can_unlock key1 lock1)*
- *(knows bob root_password)*
- *(accessible s_iexplore sherpa)*
- *(can_read_email ms_outlook)*
- *(trusts_instructions greg adam)*

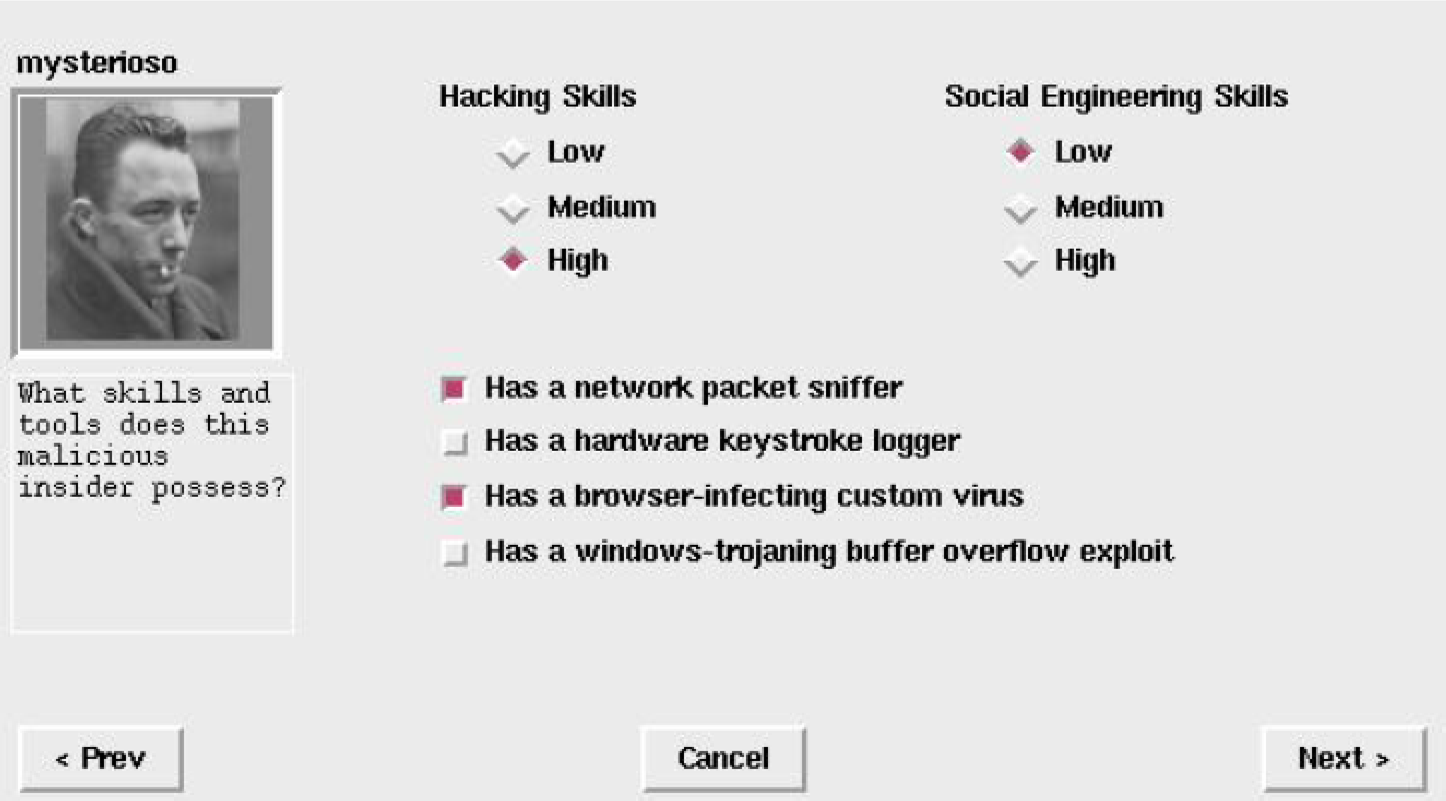
Examples: Goals

```
(:goal (knows bob secret_info))
(:metric minimize (detection_risk))

(and (knows bob secret_info)
        (<= (detection_risk) 5))
```



Adventium Labs



mysterioso

What skills and tools does this malicious insider possess?

Hacking Skills

◇ Low
◇ Medium
◆ High

Social Engineering Skills

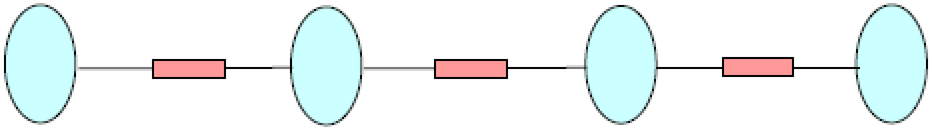
◆ Low
◇ Medium
◇ High

☒ Has a network packet sniffer
□ Has a hardware keystroke logger
☒ Has a browser-infecting custom virus
□ Has a windows-trojaning buffer overflow exploit

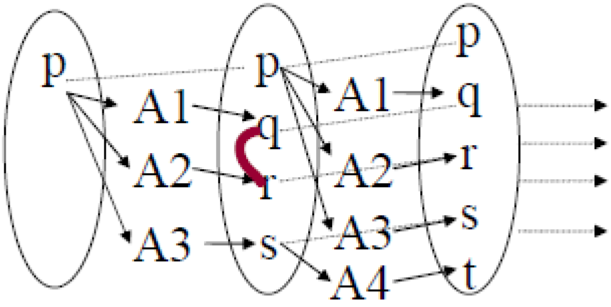
Examples: Actions

```
(action DMS_ADD_GROUP_ALLOW
    :parameters (?admin - c_human
                 ?chost - c_host
                   ?shost - c_host
                   ?doc - c_file
                   ?gid - c_gid)
    :precondition
      (and (nes_admin_connected ?chost ?shost)
           (at_host ?admin ?chost)
           (insider ?admin)
    :effect (and (dmsacl_read ?doc ?gid)))
```

Planning Graphs



Traditional state-action planning



Planning Graph

© Adventium Labs, 2005

Forward Heuristic Search

- *Hoffmann's metric FF planner*
  - *Enhanced hill climbing (EHS)*
  - *Breadth first search (BFS)*
- *Ignores mutexes*
- *Very effective for many domains*

A Plan

0 : ADAM sits down at BIGFOOT

1 : ADAM enters ADAM_UID as user name for login on host BIGFOOT

2 : ADAM enters password ADAM_PWD for login at host BIGFOOT

3 : Shell B_WEXPLORE is launched on host BIGFOOT for user ADAM_UID

4 : Program WEXPLORER on host BIGFOOT forks a child process

5 : Contents of file B_IEXPLORE begin executing as uid ADAM_UID on host BIGFOOT

6 : BOB sits down at YETI

7 : BOB enters BOB_UID as user name for login on host YETI

8 : BOB enters password BOB_PWD for login at host YETI

9 : Shell Y_WEXPLORE is launched on host YETI for user BOB_UID

10 : Program WEXPLORER on host YETI forks a child process

11 : Contents of file Y_ETHEREAL begin executing as uid BOB_UID on host YETI

12 : ETHEREAL starts sniffing the networks on YETI

13 : ADAM logs onto dms admin server EVEREST from BIGFOOT

14 : BOB reads the sniffer thus learning NES_ADMIN_PASS

Plan, Continued

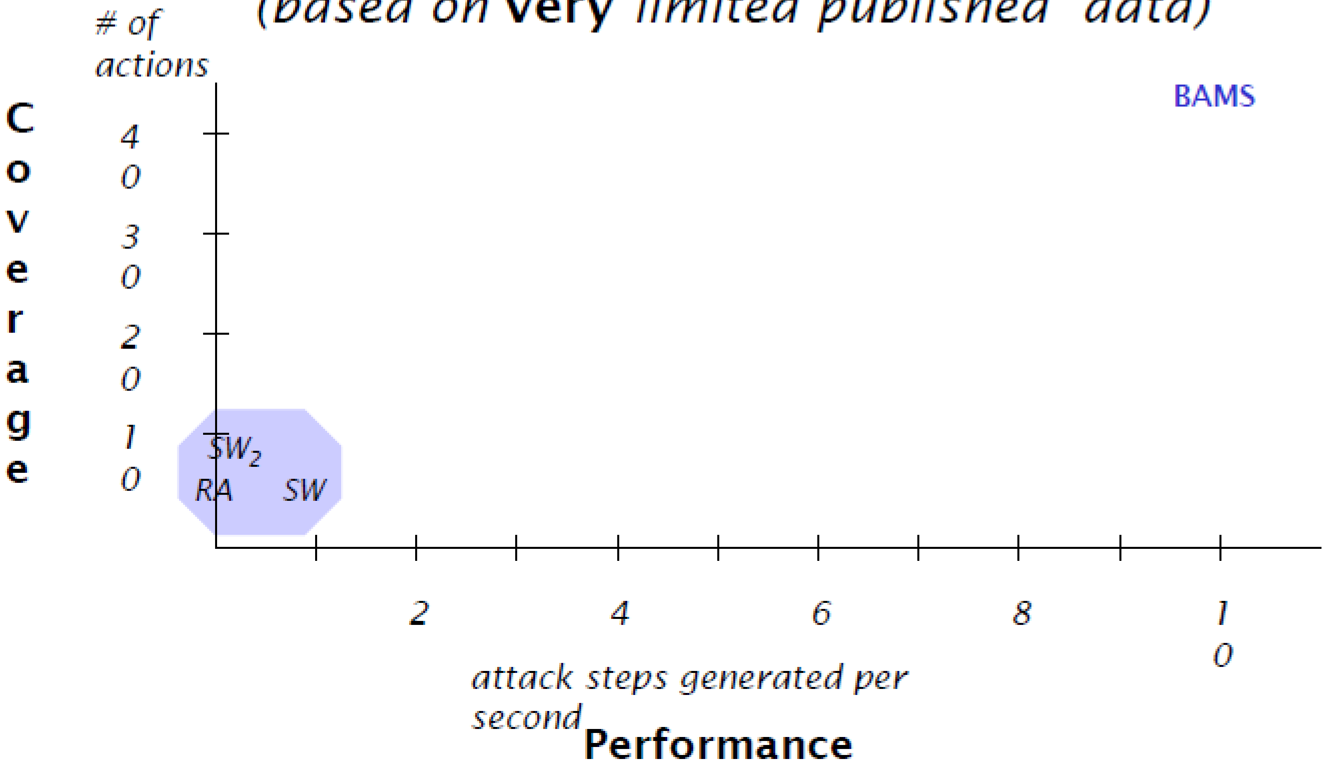
15 : Program WEXPLORER on host YETI forks a child process
16 : Contents of file Y_IEXPLORE begin executing as uid BOB_UID on host YETI
17 : BOB logs onto dms admin server EVEREST from YETI
18 : DMS session DMSS1 has begun
19 : BOB begins a DMS session on YETI
20 : Connect DMS session DMSS1 to server NES on EVEREST
21 : A route from YETI to DMS server EVEREST exists
22 : BOB enters password BOB_DMS_PWD for the DMS session.
23 : Authenticate BOB_UID in dms session DMSS1 with EVEREST using BOB_DMS_PWD
24 : BOB adds an acl to allow read access of E_SECRET_DOC to the EAST_GID group
25 : BOB begins a DMS request at YETI in session DMSS1
26 : Document E_SECRET_DOC is requested in session DMSS1
27 : Document E_SECRET_DOC is sent and displayed on YETI in session DMSS1
28 : BOB reads E_SECRET_DOC and learns SECRET_INFO

Generating Plans

	Steps	Time
Direct Client Hack	25	0.67
Misdirected Email	32	0.67
Shoulder Surfing	18	0.69
Email Trojan	37	0.71
Spoofed Email Trojan	37	0.73
Spoofed Instructions	36	0.79
Administrator ACL Change	23	1.20
Sniff Administrator Password	28	1.62
Sniff Password from Email	44	4.77

BAMS vs. Other Approaches

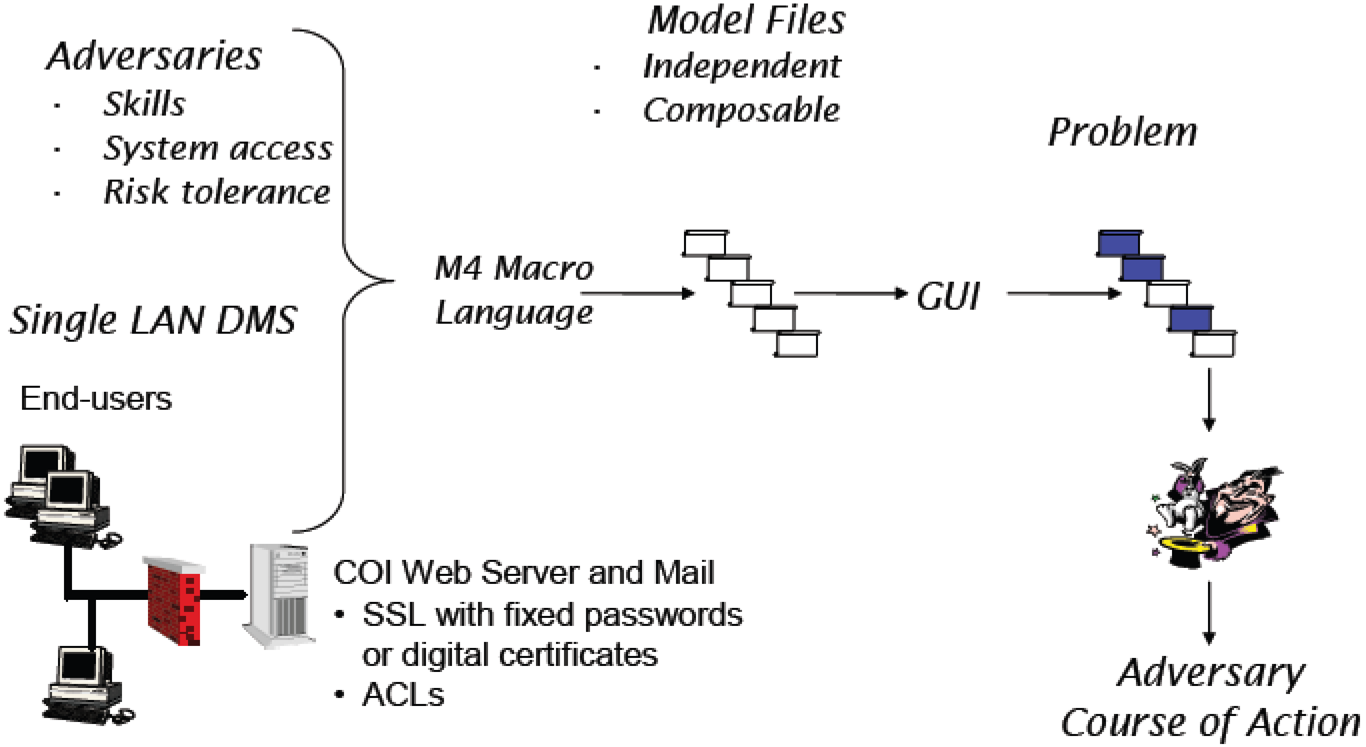
(based on **very** *limited published data)*



Pragmatic Issues

- *Performance (esp. memory consumption)*
  - *Optimizing grad-ware*
  - *Rewriting the model to avoid "hard actions"*
  - *Rewriting to minimize the size of the propositional expansion*
- *Representing processes (e.g., composing and sending email).*
- *Entities that are created or destroyed*
- *Derived predicates*
- *Maintaining large domain models*

Process



Adversaries

- *Skills*
- *System access*
- *Risk tolerance*

Single LAN DMS

End-users

COI Web Server and Mail

- SSL with fixed passwords or digital certificates
- ACLs

Sys Admin

- Password protected account
- Manages user accounts

M4 Macro Language

Model Files

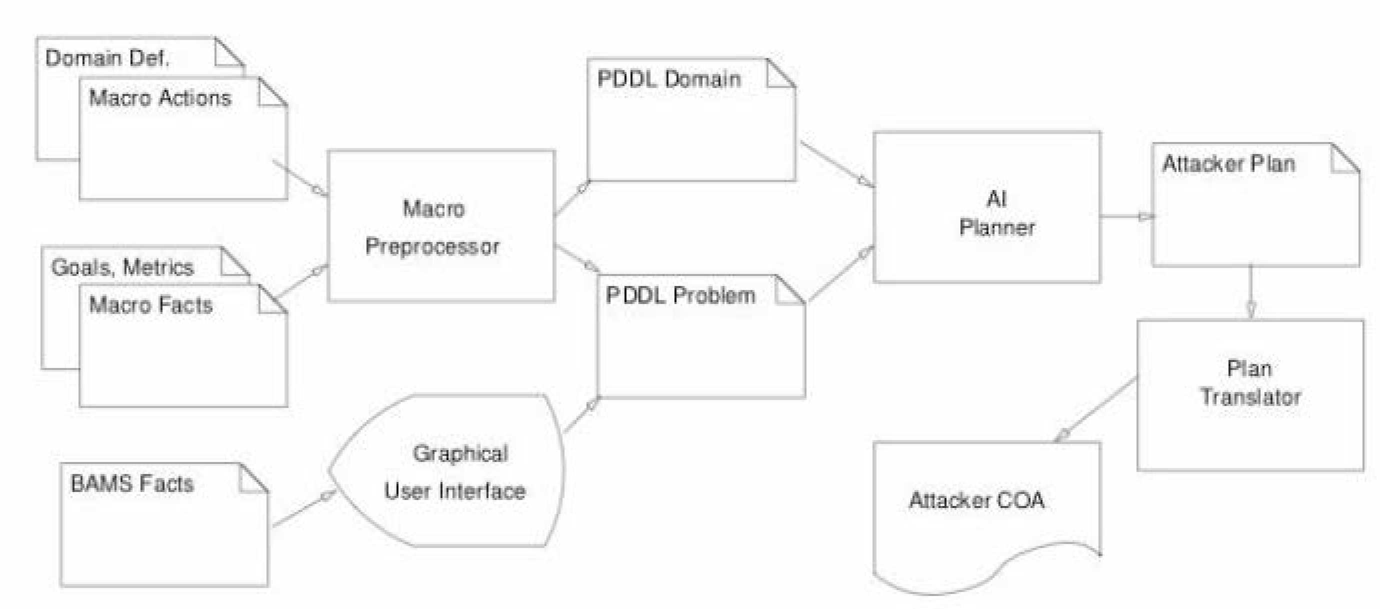
- *Independent*
- *Composable*

GUI

Problem

Adversary Course of Action

Information Flows



Future Work

- *Planner Technology*
  - *Efficient generation of multiple plans*
  - *Improvements in performance and scalability, including more extensive use of metrics*
- *Modeling Tools and Techniques*
  - *Make it easier for domain experts to extend and maintain the model*
  - *Compile user model into performance-tuned PDDL*
- *Analytic Capabilities*
  - *Bottleneck analysis*
  - *Probabilistic or uncertain reasoning*
- *IC Specific Models*
  - *Drives the work in the first three areas*
- *Comparative analysis*
  - *Head-to-head*
  - *Planning Competition*