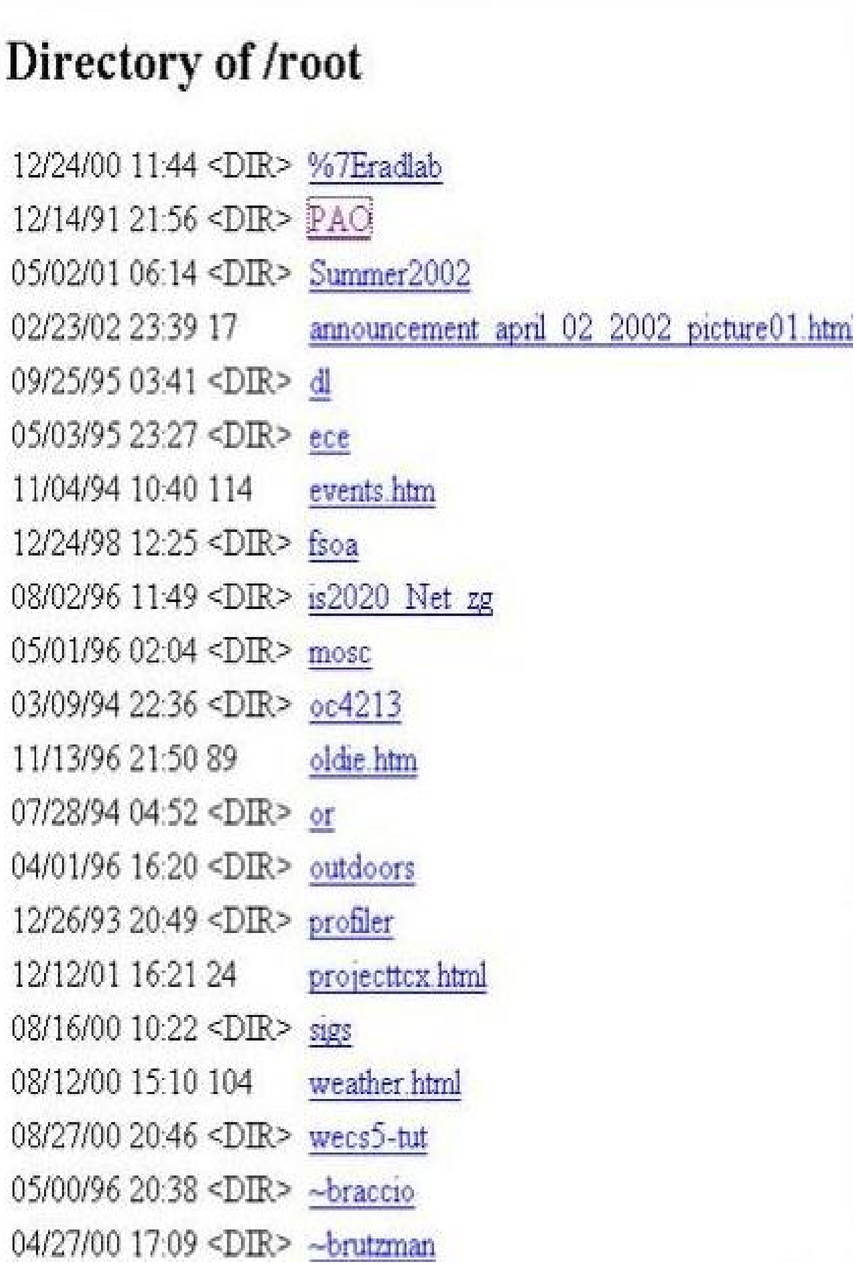
Research on deception in defense of information systems

Neil C. Rowe, Mikhail Auguston,
Doron Drusinsky, and J. Bret Michael
U.S. Naval Postgraduate School
ncrowe@nps.edu

The Software Decoy Project

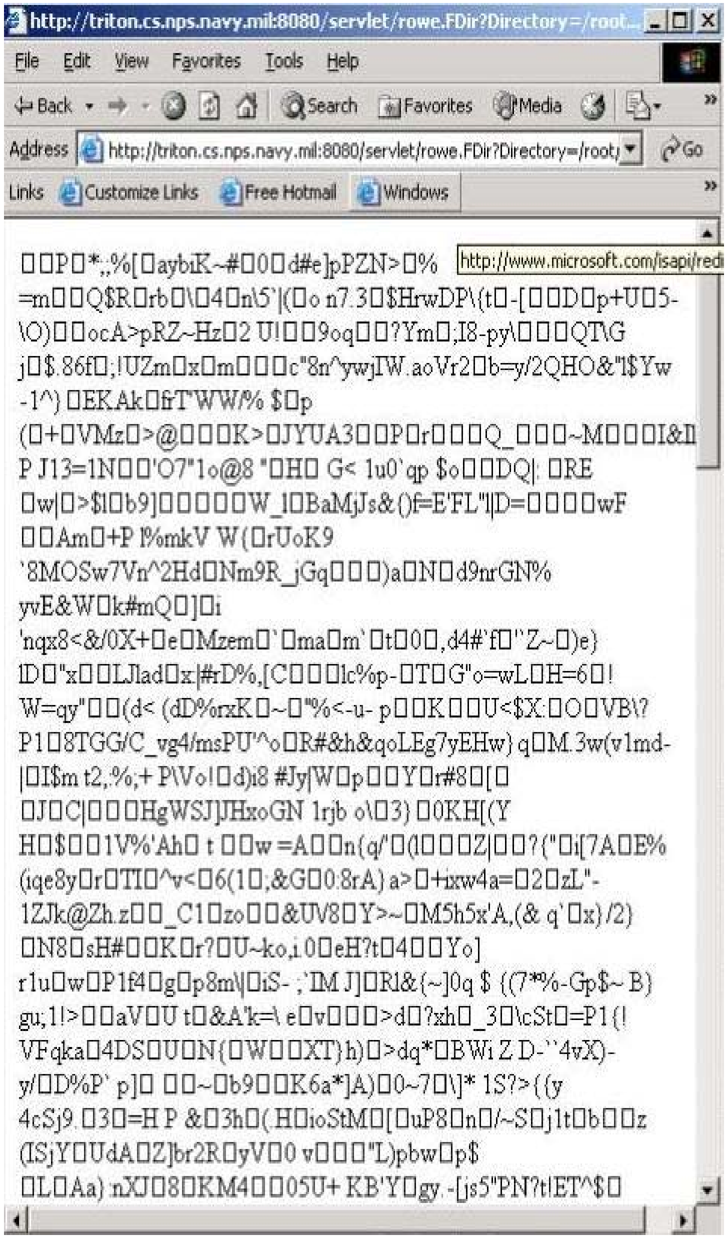
- Funded by the Department of Homeland Security and other agencies
- Tailoring classic military deception for defending computer assets
- Goal is to provide a second line of defense beyond access controls, which is especially useful for insider attacks
- Key parts of the project:
  - Theory of deception for information systems
  - Simple testbed deceptive software and ways to attach it to operating systems and applications
  - Temporal reasoning in deception
  - Legal issues in software deception

Example: A fake-directory interface



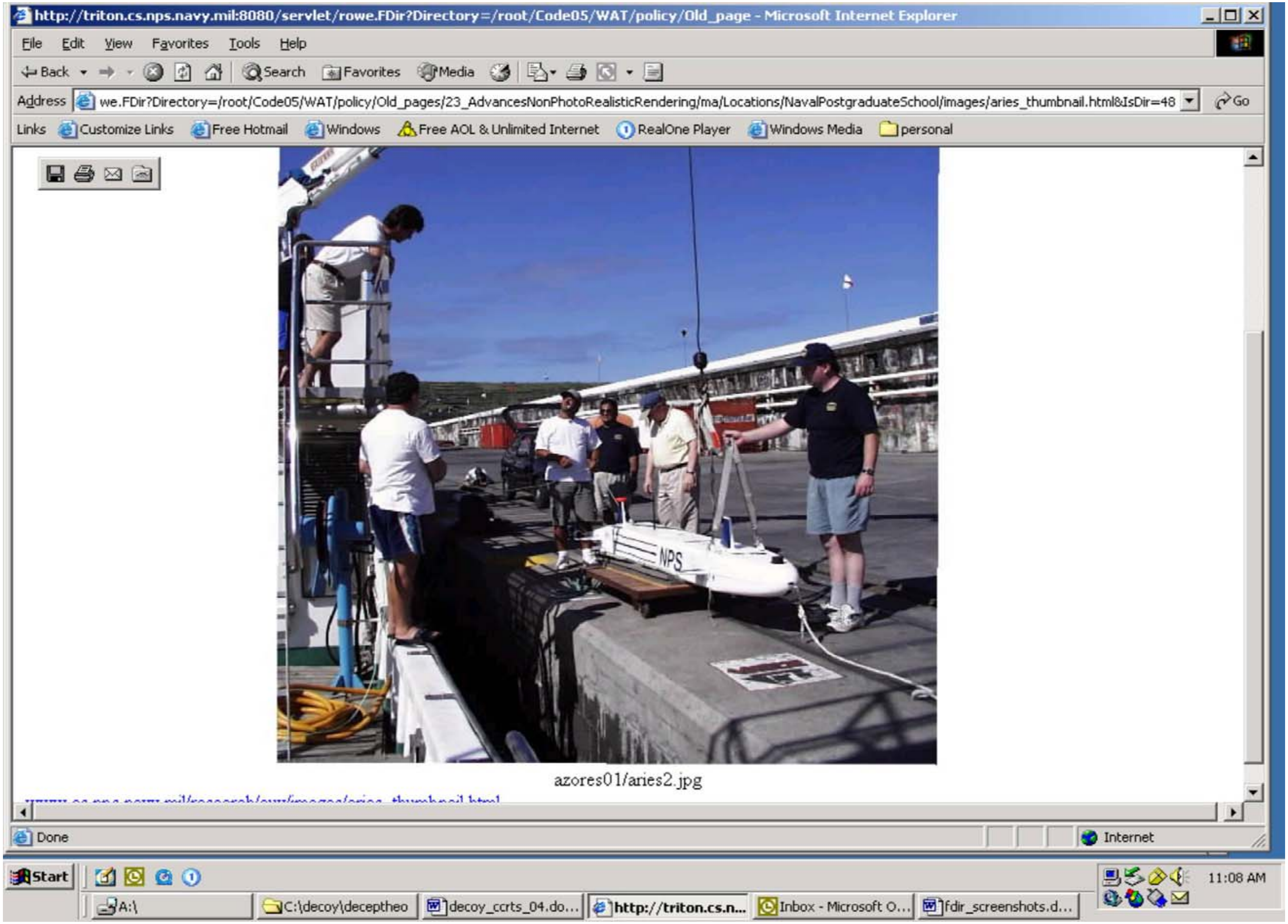
Directory of /root

```
12/24/00 11:44 <DIR> %7Eradlab
12/14/91 21:56 <DIR> PAO
05/02/01 06:14 <DIR> Summer2002
02/23/02 23:39 17    announcement_april_02_2002_picture01.html
09/25/95 03:41 <DIR> dl
05/03/95 23:27 <DIR> ece
11/04/94 10:40 114   events.htm
12/24/98 12:25 <DIR> fsoa
08/02/96 11:49 <DIR> is2020_Net_zg
05/01/96 02:04 <DIR> mosc
03/09/94 22:36 <DIR> oc4213
11/13/96 21:50 89    oldie.htm
07/28/94 04:52 <DIR> or
04/01/96 16:20 <DIR> outdoors
12/26/93 20:49 <DIR> profiler
12/12/01 16:21 24    projecttcx.html
08/16/00 10:22 <DIR> sigs
08/12/00 15:10 104   weather.html
08/27/00 20:46 <DIR> wecs5-tut
05/00/96 20:38 <DIR> ~braccio
04/27/00 17:09 <DIR> ~brutzman
```


http://triton.cs.nps.navy.mil:8080/servlet/rowe.FDir?Directory=/root...
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
http://triton.cs.nps.navy.mil:8080/servlet/rowe.FDir?Directory=/root/
Go
Links
Customize Links
Free Hotmail
Windows
http://www.microsoft.com/isapi/redi

```
□□P□*;;%[□aybiK~#□0□d#e]pPZN>□%
=m□□Q$R□rb□\□4□n\5`|(□o n7.3□$HrwDP\{t□-[□□D□p+U□5-
\O)□□ocA>pRZ~Hz□2 U!□□9oq□□?Ym□;I8-py\□□□QT\G
j□$.86f□;!UZm□x□m□□□c"8n^ywjTW.aoVr2□b=y/2QHO&"l$Yw
-1^) □EKAk□frT'WW/% $□p
(□+□VMz□>@□□□K>□JYUA3□□P□r□□□Q_□□□~M□□□I&□
P J13=1N□□'O7"1o@8 "□H□ G< 1u0`qp $o□□DQ|: □RE
□w|□>$l□b9]□□□□□W_1□BaMjJs&()f=E'FL"l|D=□□□□wF
□□Am□+P l%mkV W(□rUoK9
`8MOSw7Vn^2Hd□Nm9R_jGq□□□)a□N□d9nrGN%
yvE&W□k#mQ□]□i
'nqx8<&/0X+□e□Mzem□`□ma□m`□t□0□,d4#`f□''Z~□)e}
lD□"x□□LJlad□x|#rD%,[C□□□lc%p-□T□G"o=wL□H=6□!
W=qy"□□(d< (dD%rxK□~□"%<-u- p□□K□□U<$X:□O□VB\?
P1□8TGG/C_vg4/msPU'^o□R#&h&qoLEg7yEHw} q□M.3w(v1md-
|□I$m t2,;%;+ P\Vo!□d)i8 #Jy|W□p□□Y□r#8□[□
□J□C|□□□HgWSJ]JHxoGN 1rjb o\□3) □0KH[(Y
H□$□□1V%'Ah□ t □□w =A□□n(q/'□(l□□□Z|□□?("□i[7A□E%
(iqe8y□r□TI□^v<□6(1□;&G□0:8rA) a>□+ixw4a=□2□zL"-
1ZJk@Zh.z□□_C1□zo□□&UV8□Y>~□M5h5x'A,(& q'□x}/2)
□N8□sH#□□K□r?□U~ko,i.0□eH?t□4□□Yo]
r1u□w□P1f4□g□p8m\|□iS- ;`IM J]□Rl&{~]0q $ {(7*%-Gp$~ B)
gu;1!>□□aV□U t□&A'k=\ e□v□□□>d□?xh□_3□\cSt□=P1{!
VFqka□4DS□U□N{□W□□XT}h)□>dq*□BWi Z D-``4vX)-
y/□D%P` p]□ □□~□b9□□K6a*]A)□0~7□\]* 1S?>{(y
4cSj9.□3□=H P &□3h□(.H□ioStM□[□uP8□n□/~S□j1t□b□□z
(ISjY□UdA□Z]br2R□yV□0 v□□□"L)pbw□p$
□L□Aa) nXJ□8□KM4□□05U+ KB'Y□gy.-[js5"PN?t!ET^$□
```

Example random-context data from the fake directory



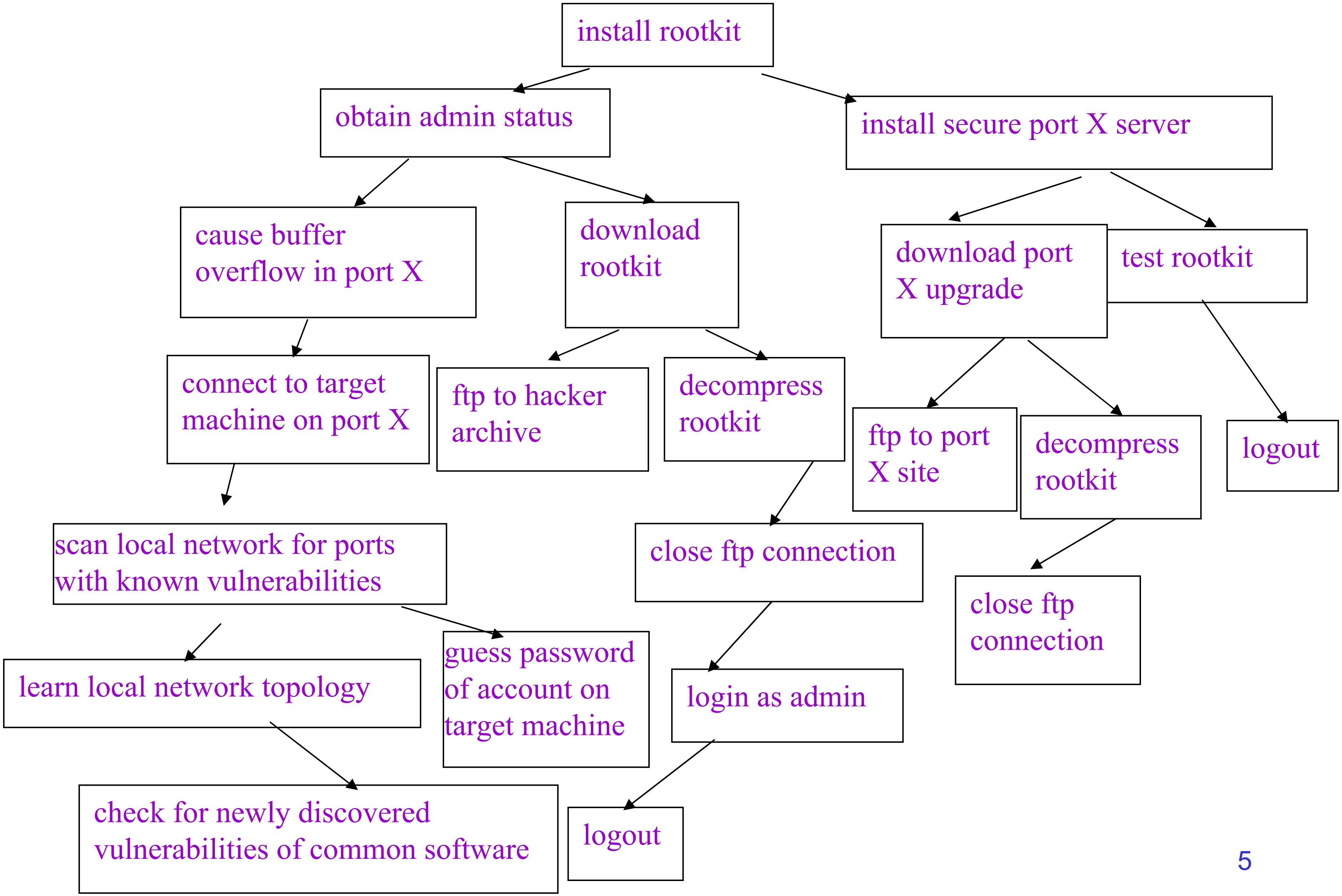
http://triton.cs.nps.navy.mil:8080/servlet/rowe.FDir?Directory=/root/Code05/WAT/policy/Old_page - Microsoft Internet Explorer

Address: we.FDir?Directory=/root/Code05/WAT/policy/Old_pages/23_AdvancesNonPhotoRealisticRendering/ma/Locations/NavalPostgraduateSchool/images/aries_thumbnail.html&IsDir=48

azores01/aries2.jpg

www.cs.nps.navy.mil/research/auv/images/aries_thumbnail.html

Example simple cyber-attack plan for a rootkit



install rootkit

obtain admin status

install secure port X server

cause buffer overflow in port X

download rootkit

download port X upgrade

test rootkit

connect to target machine on port X

ftp to hacker archive

decompress rootkit

ftp to port X site

decompress rootkit

logout

scan local network for ports with known vulnerabilities

close ftp connection

close ftp connection

learn local network topology

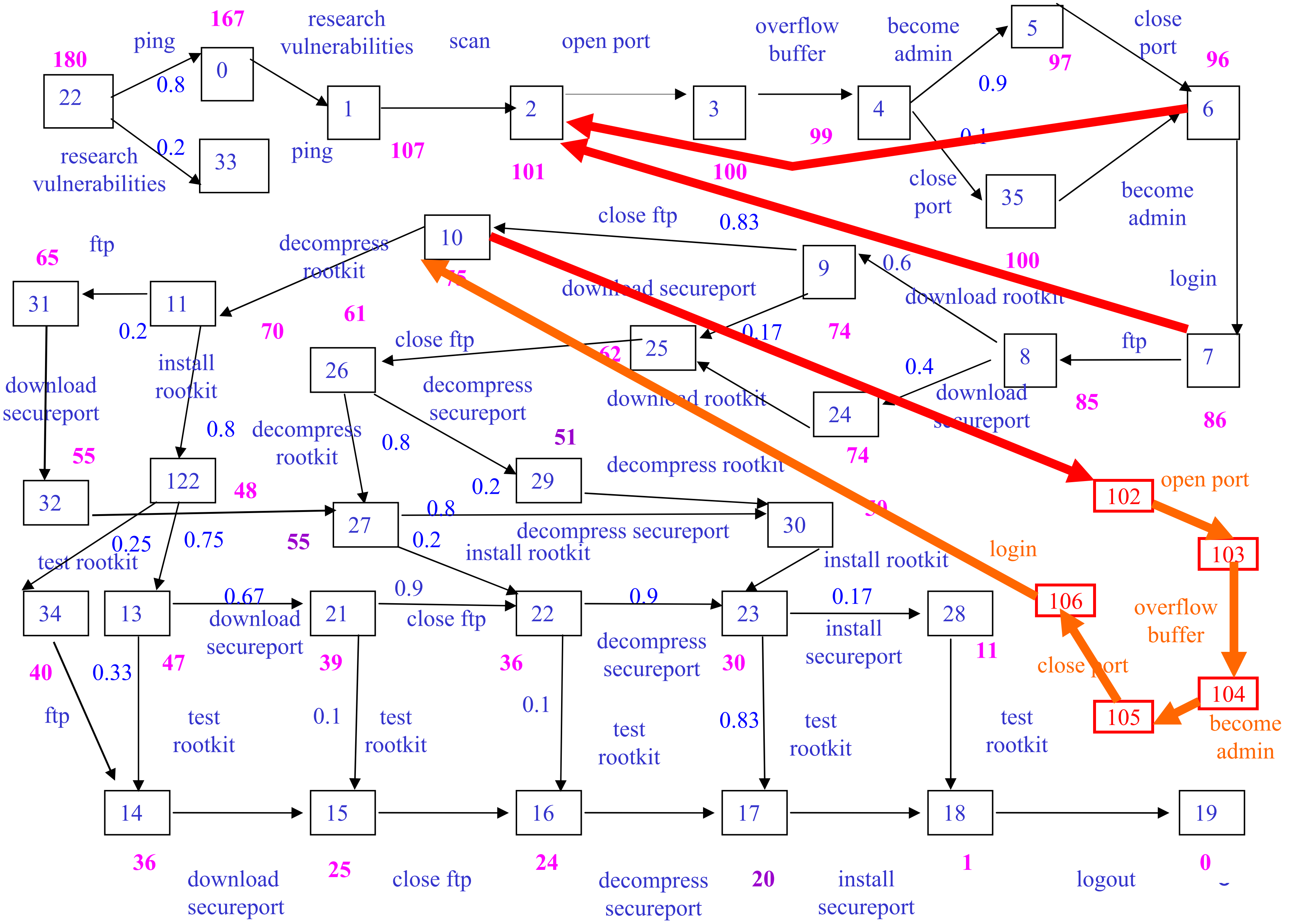
guess password of account on target machine

login as admin

check for newly discovered vulnerabilities of common software

logout

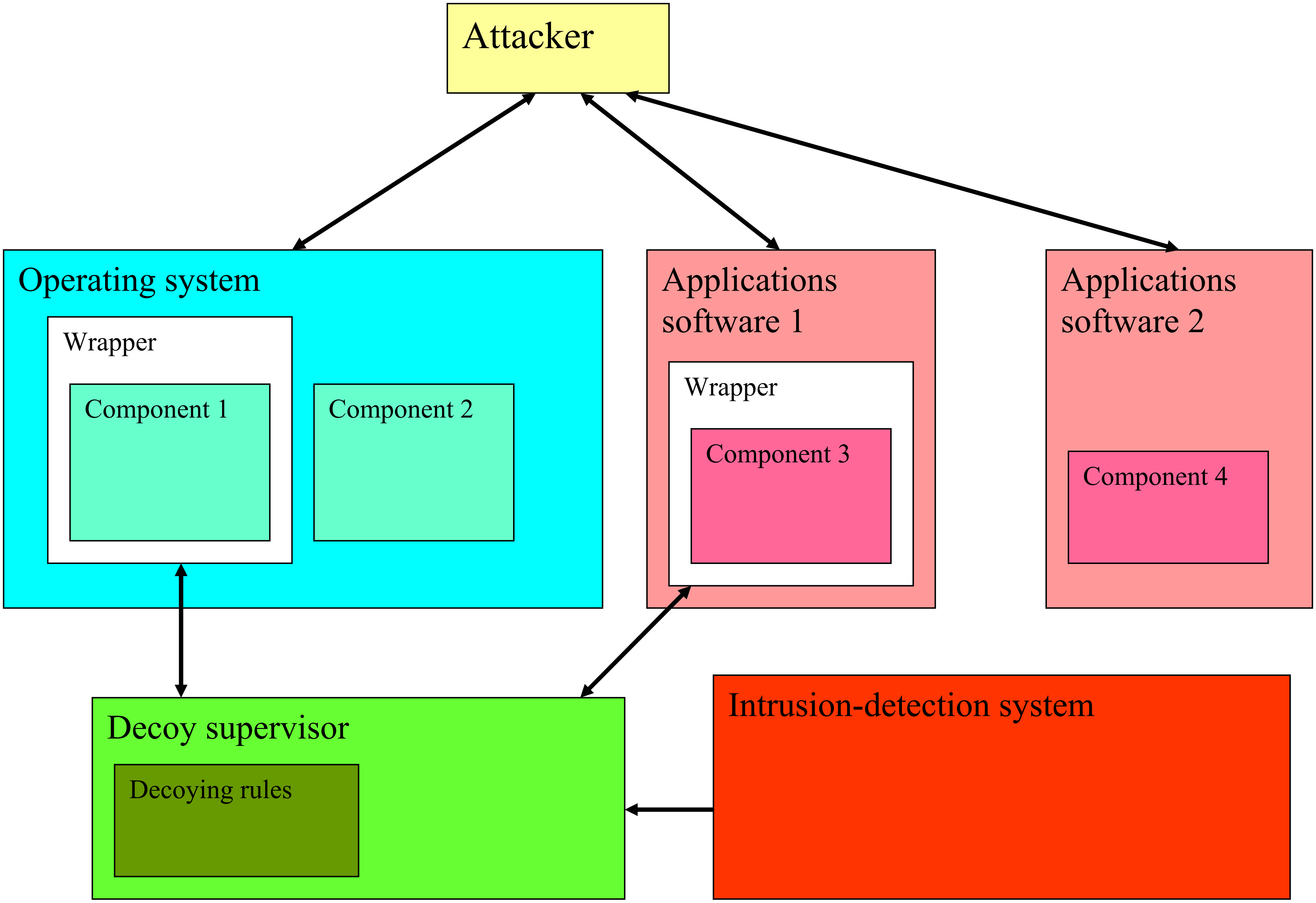
Example ploy: Delete admin authorization + log out



Software wrappers for deception

- For a deception defense to be effective, it is good to distribute it across many features of an operating system -- like "antibodies".
- We are building tools to automatically modify software to insert "wrappers" around key code; the wrappers can apply deception when their suspicions are aroused.

General decoy architecture



Example deception rule for a software wrapper

- This detects opening a file, read/write operations, and closing the file.
- Each event cause a message to the system log file.
- The **pre** and **post** indicate whether the action is done before or after the matching kernel call.
- $path provides values of kernel call parameters.
- Besides executing code, wrapper rules may prevent or delay execution of a kernel call.

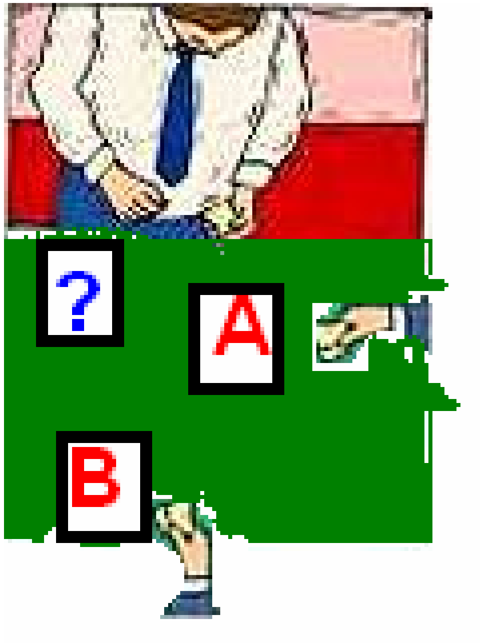
```
   R1 :  detect
open pre   { wr_printf("open file %s", $path) ; }
(          read pre { wr_printf("read file %s ", $path); }    |
  write pre {wr_printf("write file %s ", $path);}       ) *
           close post { wr_printf( "file %s  closed", $path); }
```

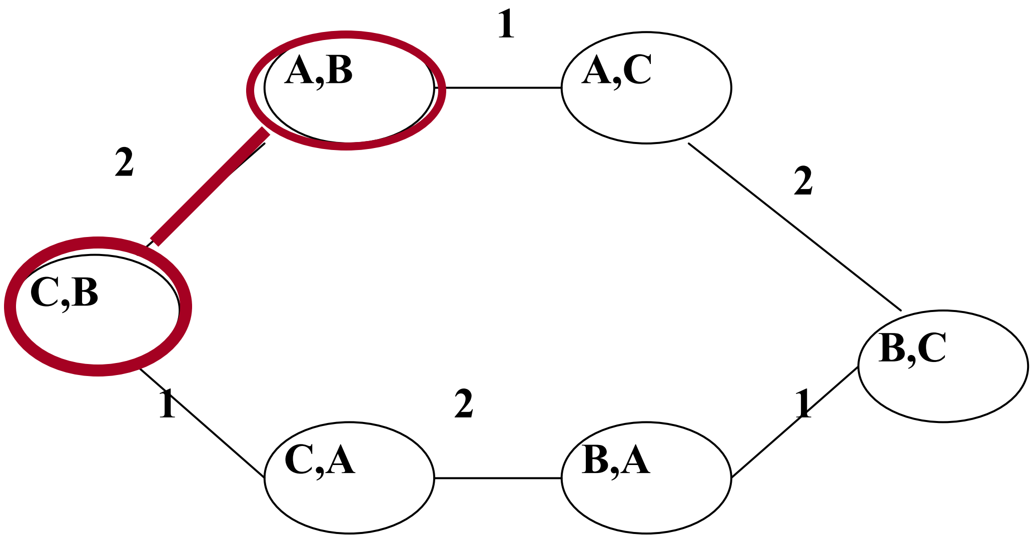
Timing in deceptions

- Deceptions involve sequences of activities in time.
- In some deceptions, the timing of these activities is critical.
- Since people have difficulty reasoning about time, it is helpful to formalize complex activities for computer analysis.
- We use "KTL", knowledge temporal logic.

Knowledge Logic

Cards Game





Possible Worlds Model

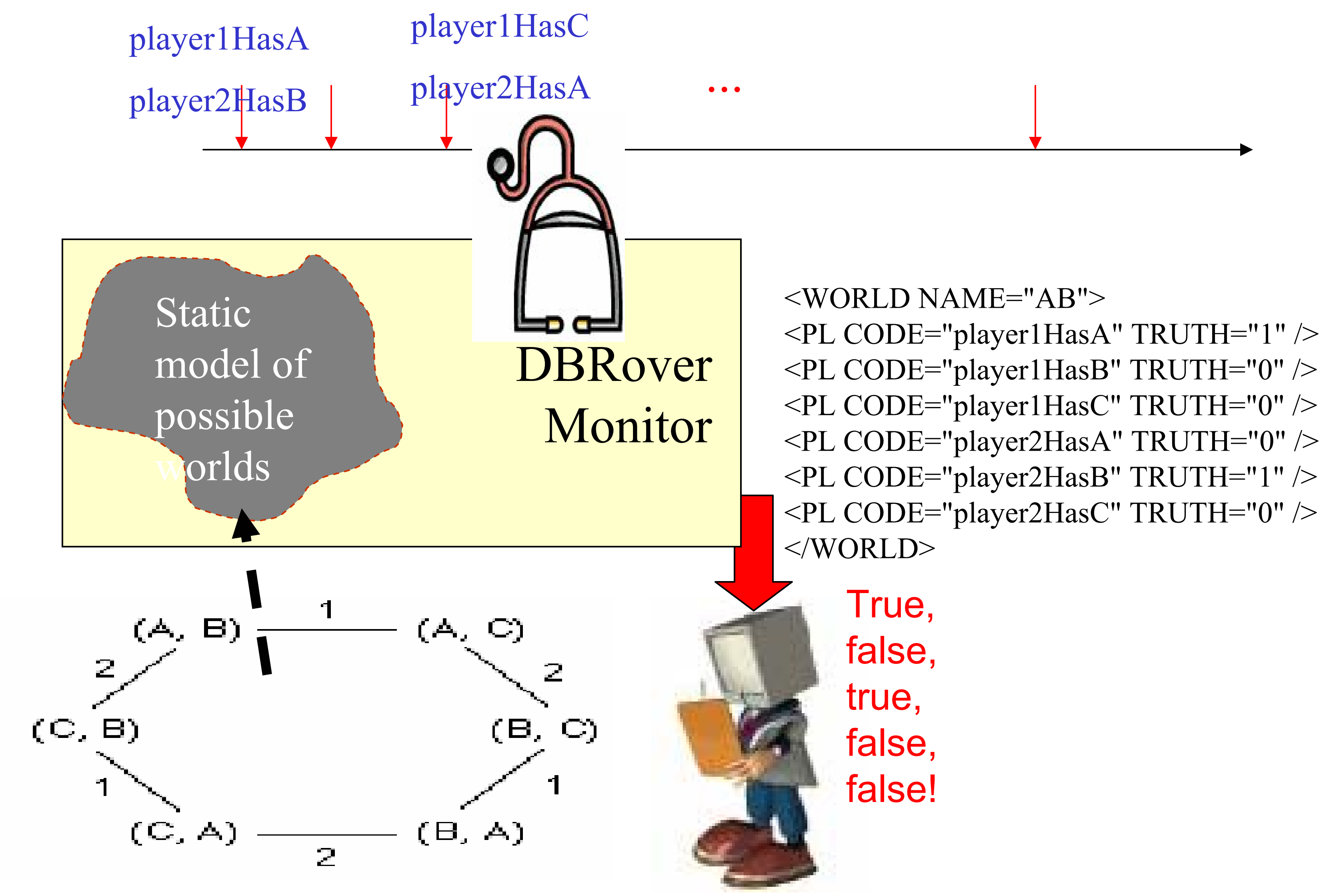
$\langle A,B\rangle \models \text{Knows}_{\text{player2}}\ \text{player1hasCardA}$

Statement must be true in **all** worlds possible for player2 when in <A,B>

False, because in C,B:

player1hasCardA = *false*

KTL: Monitoring – static possible worlds model



player1HasA

player2HasB

player1HasC

player2HasA

…

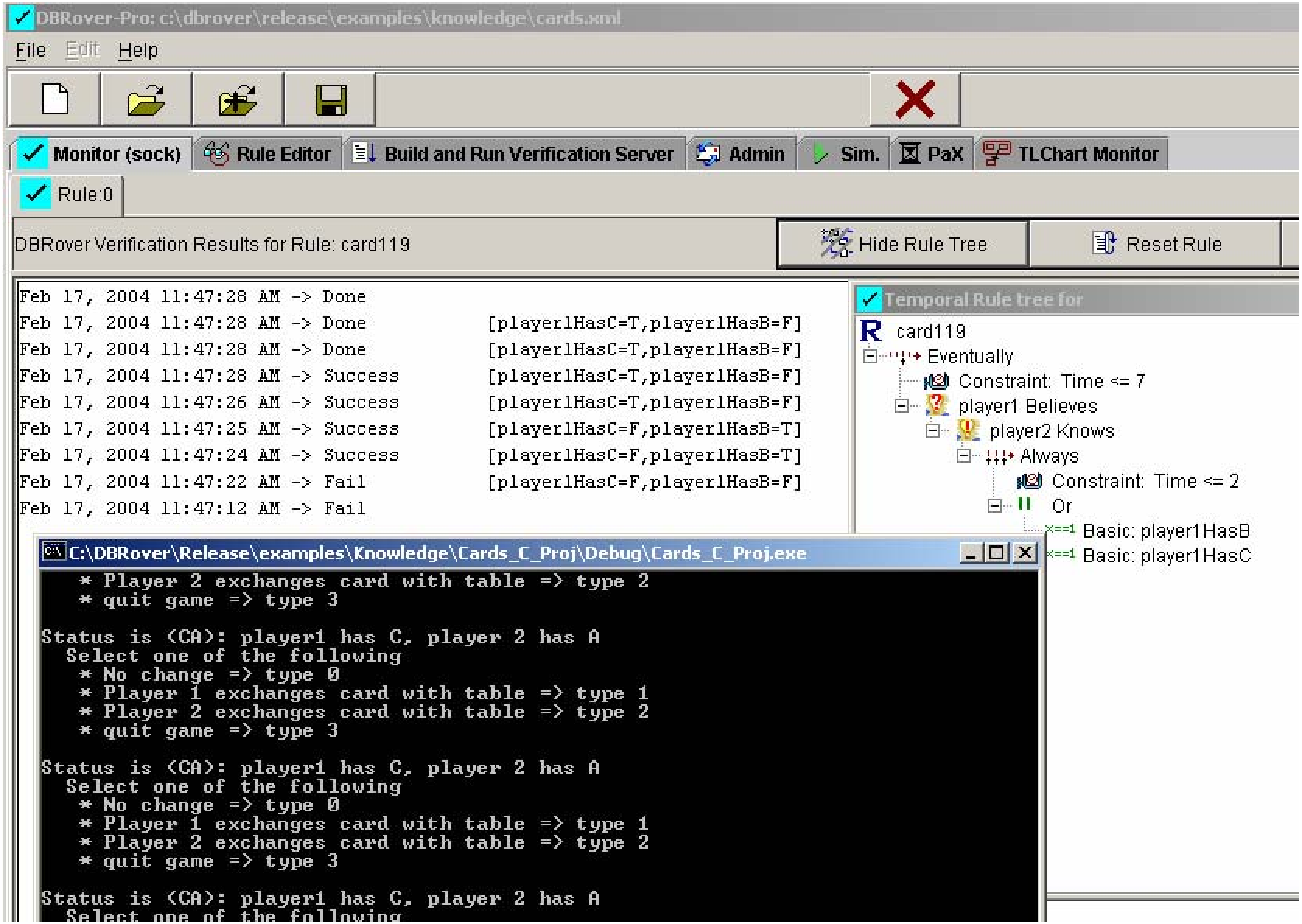
Static model of possible worlds

DBRover Monitor

```
<WORLD NAME="AB">
<PL CODE="player1HasA" TRUTH="1" />
<PL CODE="player1HasB" TRUTH="0" />
<PL CODE="player1HasC" TRUTH="0" />
<PL CODE="player2HasA" TRUTH="0" />
<PL CODE="player2HasB" TRUTH="1" />
<PL CODE="player2HasC" TRUTH="0" />
</WORLD>
```

True,
false,
true,
false,
false!

KTL: Monitoring


DBRover-Pro: c:\dbrover\release\examples\knowledge\cards.xml
File
Edit
Help
Monitor (sock)
Rule Editor
Build and Run Verification Server
Admin
Sim.
PaX
TLChart Monitor
Rule:0
Hide Rule Tree
Reset Rule
Temporal Rule tree for
C:\DBRover\Release\examples\Knowledge\Cards_C_Proj\Debug\Cards_C_Proj.exe

DBRover Verification Results for Rule: card119

```
Feb 17, 2004 11:47:28 AM -> Done
Feb 17, 2004 11:47:28 AM -> Done         [player1HasC=T,player1HasB=F]
Feb 17, 2004 11:47:28 AM -> Done         [player1HasC=T,player1HasB=F]
Feb 17, 2004 11:47:28 AM -> Success      [player1HasC=T,player1HasB=F]
Feb 17, 2004 11:47:26 AM -> Success      [player1HasC=T,player1HasB=F]
Feb 17, 2004 11:47:25 AM -> Success      [player1HasC=F,player1HasB=T]
Feb 17, 2004 11:47:24 AM -> Success      [player1HasC=F,player1HasB=T]
Feb 17, 2004 11:47:22 AM -> Fail         [player1HasC=F,player1HasB=F]
Feb 17, 2004 11:47:12 AM -> Fail
```

```
R card119
  Eventually
    Constraint:  Time <= 7
    player1 Believes
      player2 Knows
        Always
          Constraint:  Time <= 2
          Or
            x==1 Basic: player1HasB
            x==1 Basic: player1HasC
```

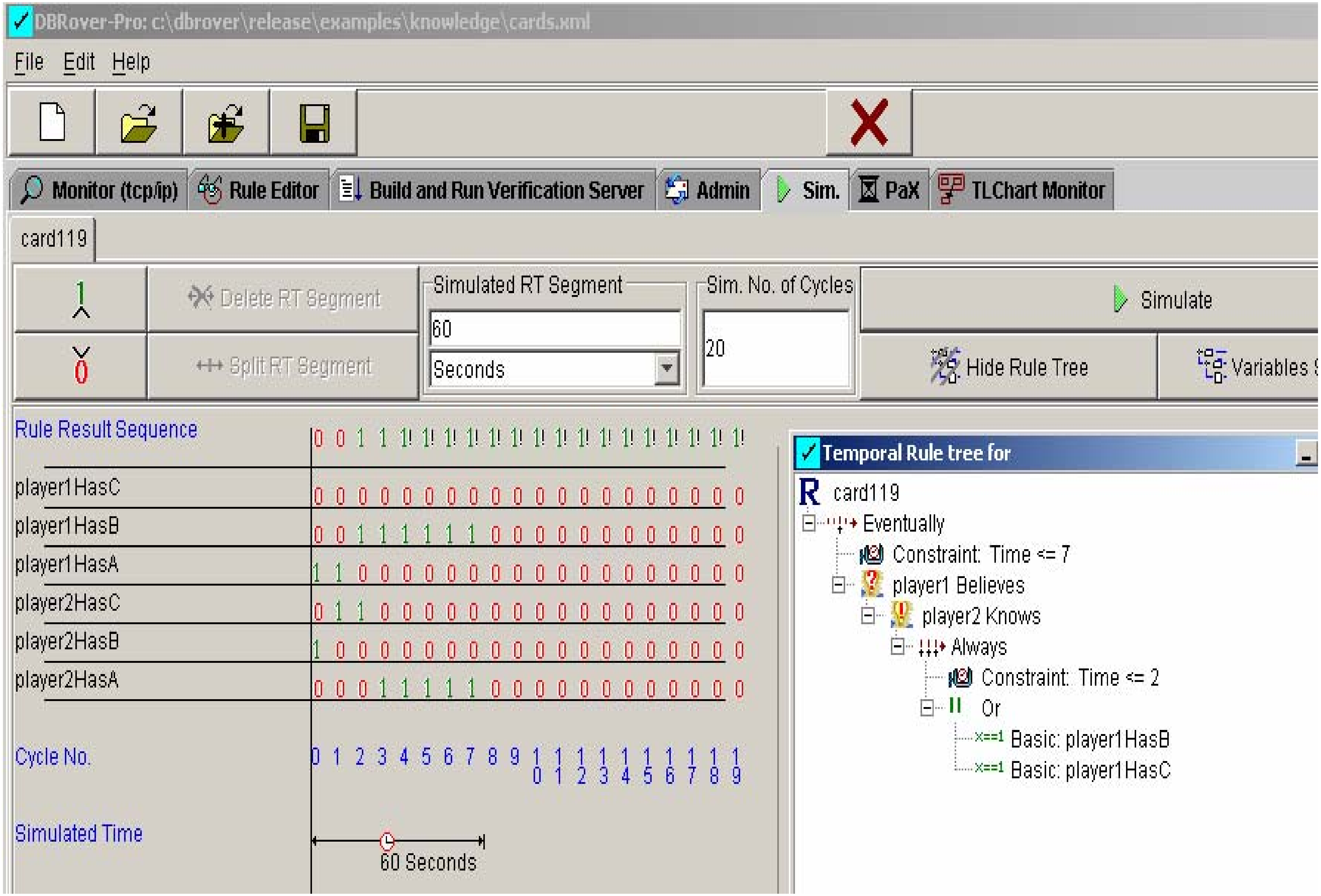
```
    * Player 2 exchanges card with table => type 2
    * quit game => type 3

Status is (CA): player1 has C, player 2 has A
   Select one of the following
    * No change => type 0
    * Player 1 exchanges card with table => type 1
    * Player 2 exchanges card with table => type 2
    * quit game => type 3

Status is (CA): player1 has C, player 2 has A
   Select one of the following
    * No change => type 0
    * Player 1 exchanges card with table => type 1
    * Player 2 exchanges card with table => type 2
    * quit game => type 3

Status is (CA): player1 has C, player 2 has A
   Select one of the following
```

KTL: Simulation



Simulated RT Segment: 60 Seconds

Sim. No. of Cycles: 20

Rule Result Sequence	0	0	1	1	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!	1!
player1HasC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
player1HasB	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
player1HasA	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
player2HasC	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
player2HasB	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
player2HasA	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Cycle No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Simulated Time: 60 Seconds

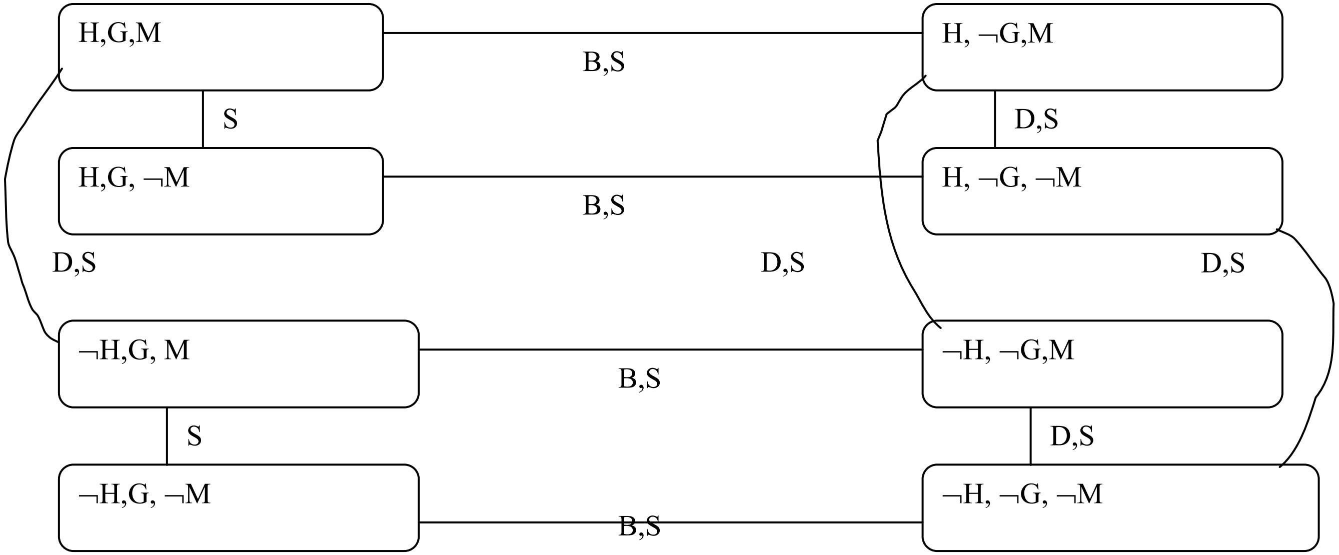
Temporal Rule tree for

- R card119
  - Eventually
    - Constraint: Time <= 7
    - player1 Believes
      - player2 Knows
        - Always
          - Constraint: Time <= 2
          - Or
            - Basic: player1HasB
            - Basic: player1HasC

Kripke model for the "Man Who Never Was"

This gives all possible worlds seen by the three agents, the British, the Germans, and the Spanish. We define the following three Boolean propositions, which together induce a space of eight possible worlds: H- represents possible worlds where Major Martin episode is a deception: G- represents possible worlds where the German coroner is in Spain and is working on the case; M- represents possible worlds where Major Martin drowned.

Hence, for example, w1 = <H, ¬G, ¬M> is the possible world where the Major Martin episode is a deception, the German coroner is not in Spain, and Major Martin did not drown. This is the possible world the British considered they were in, but in fact, they were unable to distinguish between this world and w2 = <H, G, ¬M> and could have very well been in world w2.



Legal issues in software deception

- Deception applied by a government is limited by law and policy, the former of which can be represented by mechanical rules.
- The policy (latitude with which to apply the law) is not readily amenable to full automation, but we are developing decision-support tools for assessing deception options.
- An area in which this is critical is defense against cyber-terrorism.
- We developed THEMIS, a threat evaluation "metamodel" for information systems that organizes a legal case against computer network attacks.