Extracting and Visualizing Security Properties using Self-Propelled Instrumentation

Para dyn™

UNIVERSITY OF MARYLAND
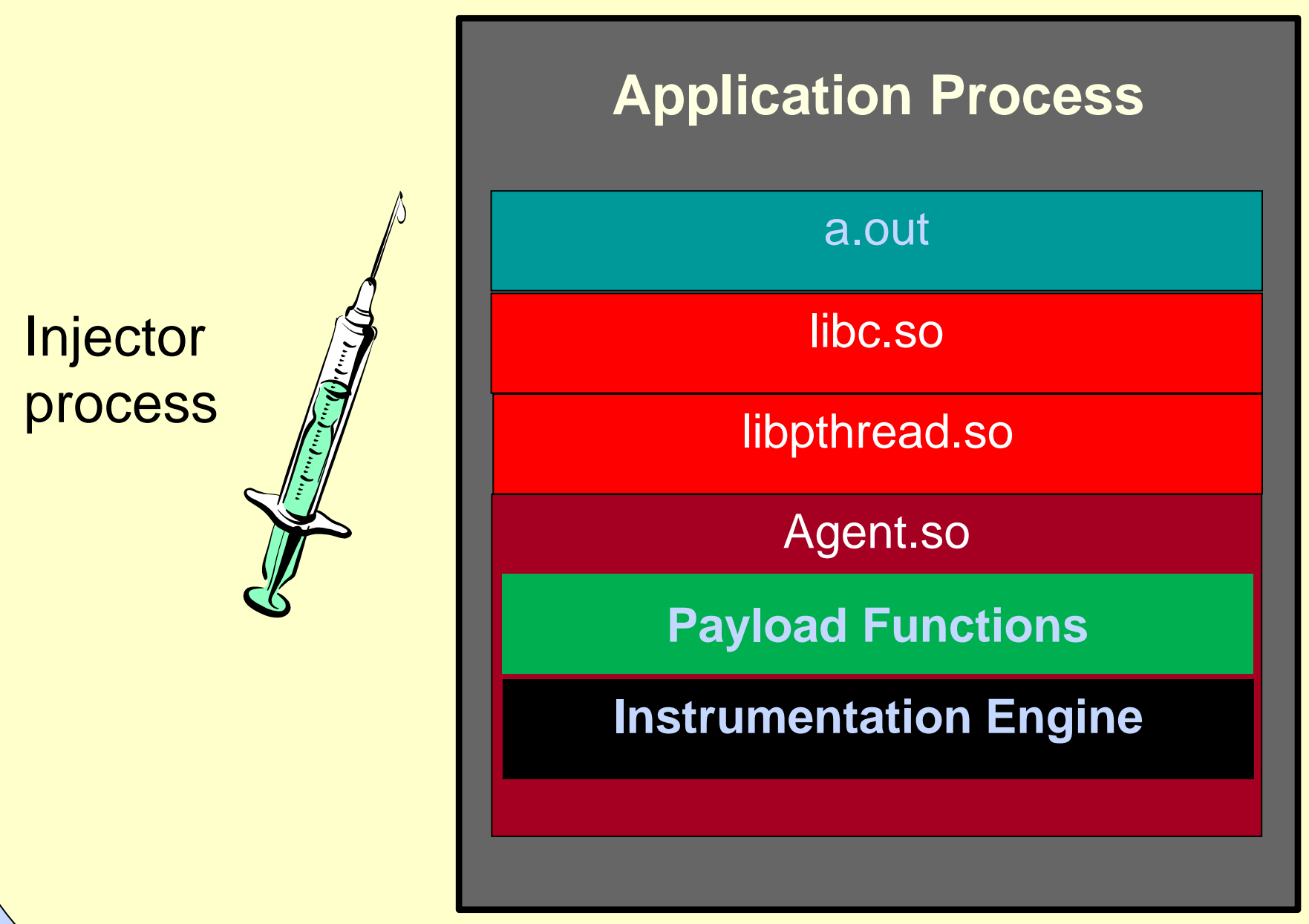
THE UNIVERSITY of WISCONSIN MADISON

Overview

➢ Inject initial instrumentation into the system at runtime

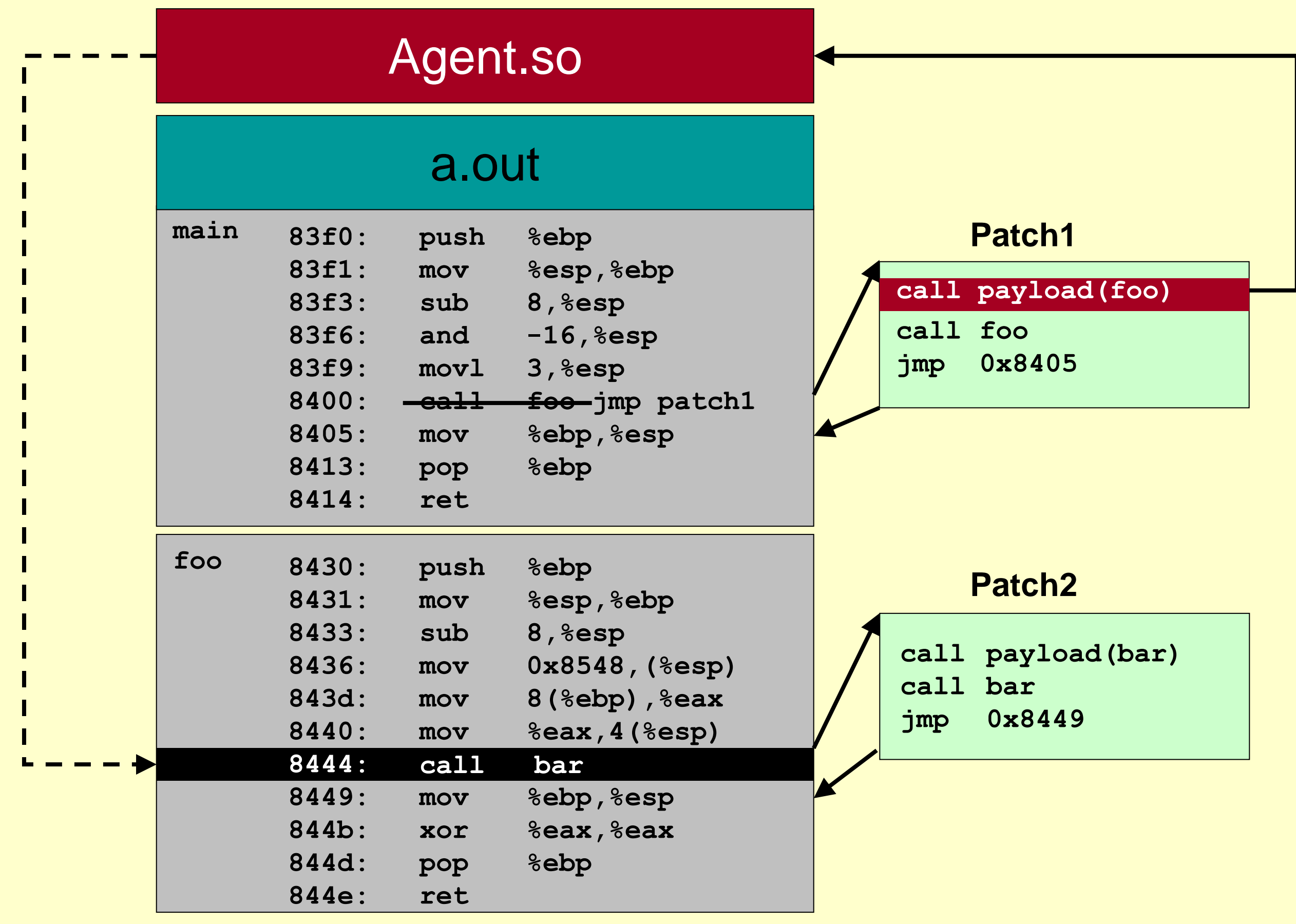➢ The instrumentation inserts more instrumentation ahead of the flow of control

Components

➢ **Injector**: forces an application progress to load a shared library

➢ **Agent:** a shared library

- Payload function: contains user-specified code
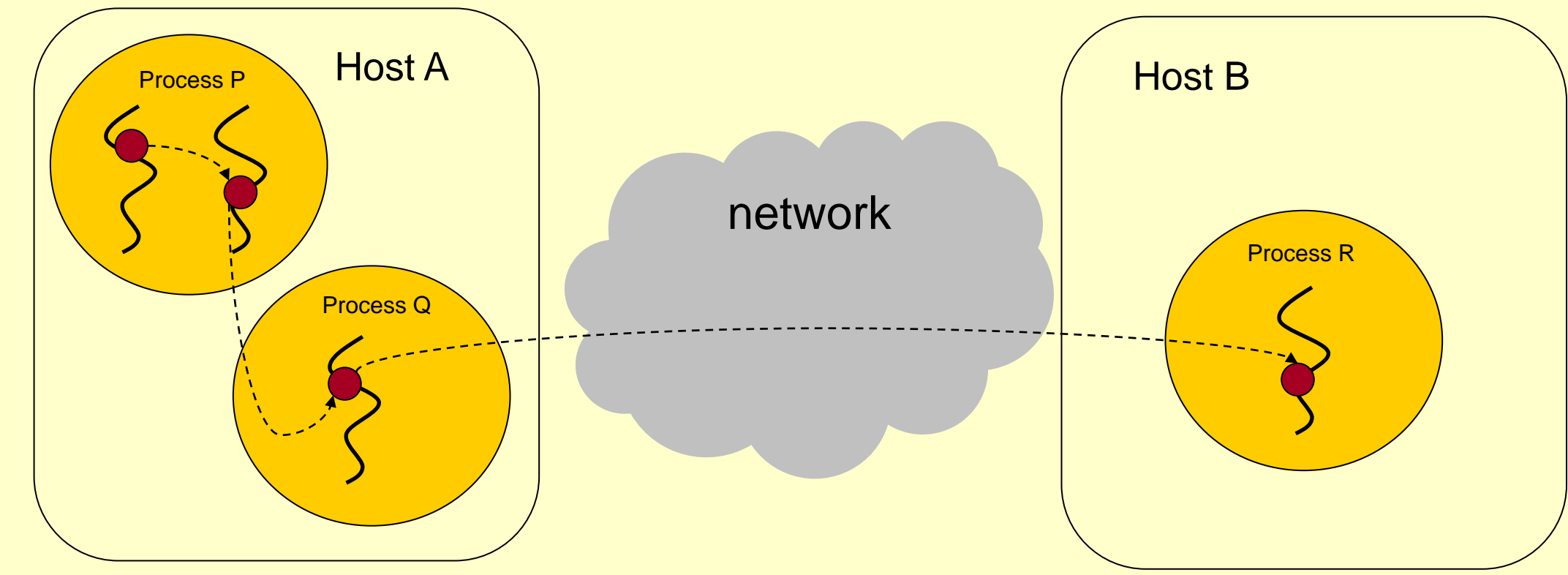
- Instrumentation engine

Injector process

Application Process

| a.out |
| libc.so |
| libpthread.so |
| Agent.so |
| Payload Functions |
| Instrumentation Engine |

Instrumentation Propagation

➢ Instrumentation propagates **_within_** a process by following control flow

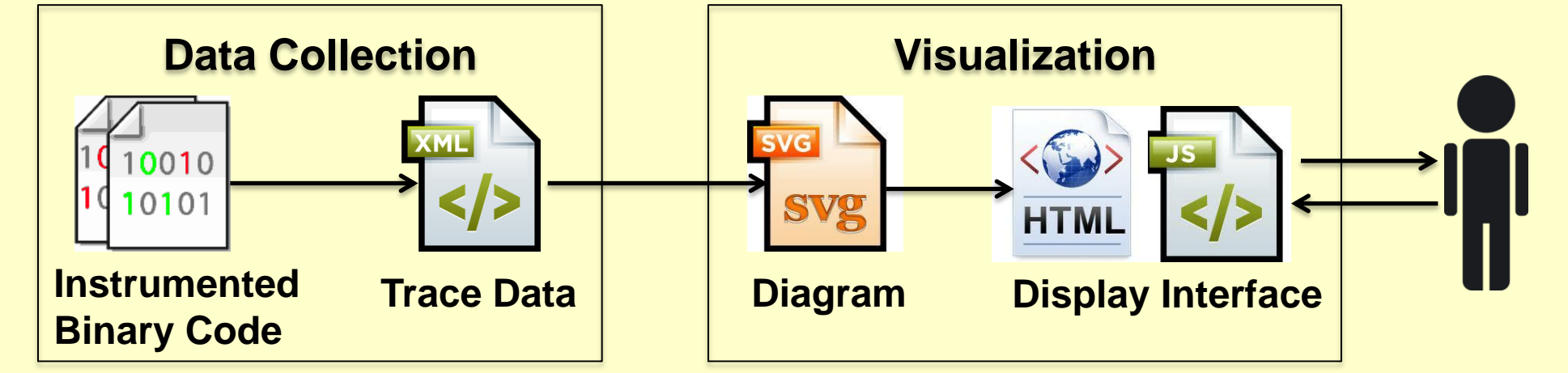**Agent.so**

a.out

```
main   83f0:   push   %ebp
       83f1:   mov    %esp,%ebp
       83f3:   sub    8,%esp
       83f6:   and    -16,%esp
       83f9:   movl   3,%esp
       8400:   call foo jmp patch1
       8405:   mov    %ebp,%esp
       8413:   pop    %ebp
       8414:   ret
```

Patch1
```
call payload(foo)
call foo
jmp  0x8405
```

```
foo    8430:   push   %ebp
       8431:   mov    %esp,%ebp
       8433:   sub    8,%esp
       8436:   mov    0x8548,(%esp)
       843d:   mov    8(%ebp),%eax
       8440:   mov    %eax,4(%esp)
       8444:   call   bar
       8449:   mov    %ebp,%esp
       844b:   xor    %eax,%eax
       844d:   pop    %ebp
       844e:   ret
```

Patch2
```
call payload(bar)
call bar
jmp  0x8449
```

➢ Instrumentation propagates **_across_** thread, process, and host boundaries by following communication flow

Host A — Process P, Process Q

network

Host B — Process R

Application: Visualization for Security Analysis

➢ Goal: automate diagram construction for security assessment

Data Collection
Instrumented Binary Code → Trace Data

Visualization
Diagram → Display Interface

➢ Data collection

- Use self-propelled instrumentation

- Detect system events and security properties

➢ Diagram construction from trace data

192.168.8.133   192.168.8.135

192.168.8.134

Legend

Abbr. for executables:
a: condor_shadow
c: condor_collector
e: condor_exec.exe
i : condor_mips
h: condor_schedd
k: condor_kflops
m: condor_master
n : condor_negotiator
p: condor_procd
s: condor_startd
r : rm
u: condor_submit
t : condor_starter

Colors for privilege level:
root   condor   wenbin

Nodes and edges:
fork
clone
connect

A process

Multiple identical processes

Host