



DETER/EMIST

Security Experimenter's Workbench

Stephen Schwab
June 16, 2006



Support Facilities

- 1. Library of security experiment building blocks**
 - Code: tools, scripts, GUI-frameworks
 - Data: packet traces, topologies and other static info that is experiment-neutral.
- 2. Repository of complete experiments**
- 3. Integrated experimenter environment and interface: Security Experimenter's Workbench**



1. Tools Library

- **First step: assemble or build a library of tools that addresses each of the canonical experiment dimensions:**
 - 1. Attack generators
 - 2. Background traffic generators
 - 3. Topology generators
 - 4. Defense configuration
 - 5. Instrumentation, measurement, data analysis and visualization
 - 6. Infrastructure configuration tools (e.g. DNS, routing)
 - 7. Risk mitigation (packet filtering predicates)



Tools Library (Examples)

- **Attack generators**
 - TCPopera (UCDavis)
 - Minos-honeypot (UC Davis)
 - BGPplay (UCDavis)
 - SPINES (UCDavis)
 - Flood – Purdue
 - Event-based agents (Sparta)
 - Wormgen (UCDavis)

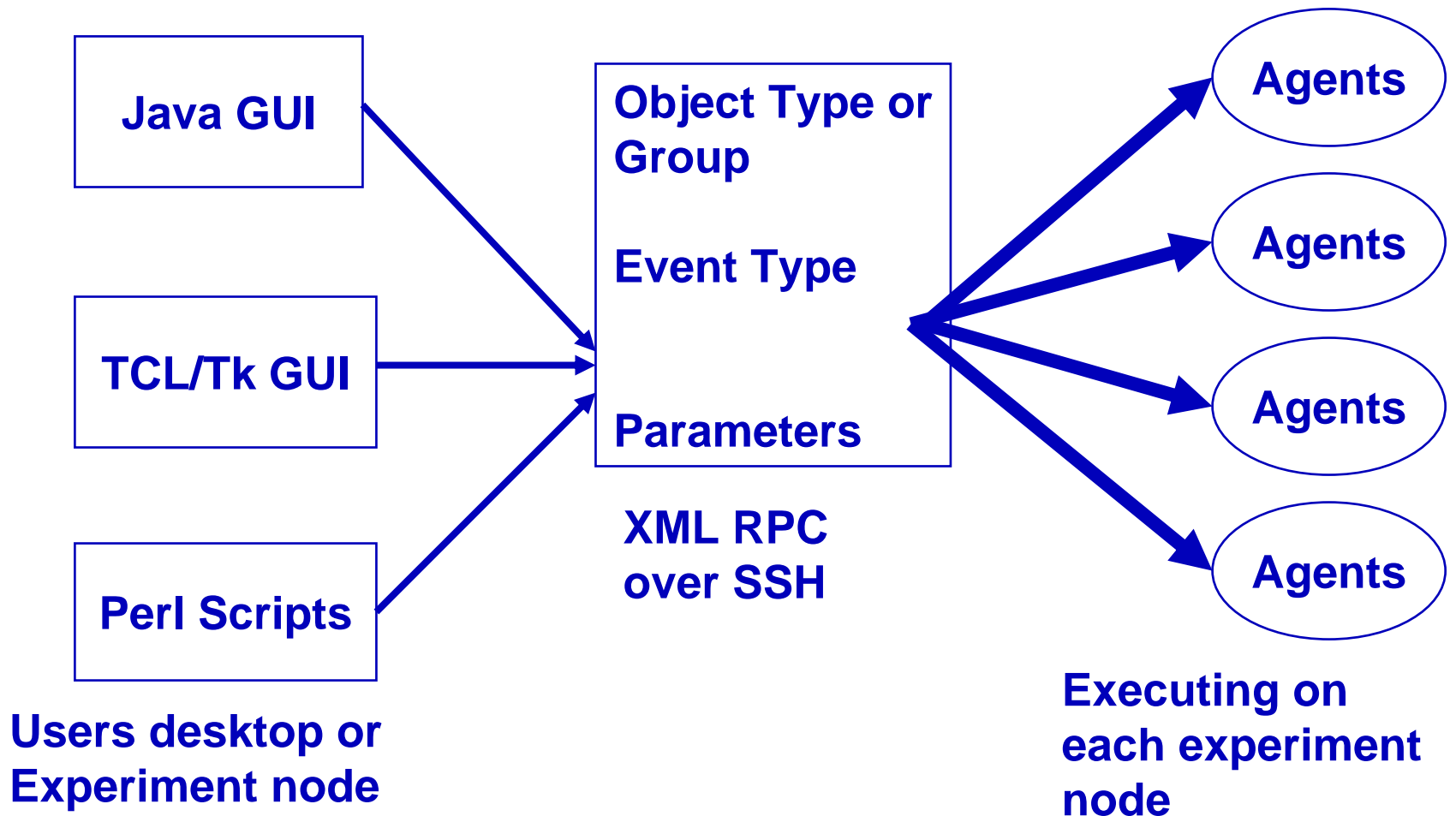
O&M for tools: (1) source and binaries, (2) tarfiles installed with experiment nodes, (3) default images on experiment nodes



2. Experiment Repository

- **Vision: an archive of experiments performed on DETER**
- **"Hello World": Simple complete experiments for new users and students**
 - Generally simple
 - Plan initial use in USC security classes
- **"FloodWatch DDoS": More complex DDoS experiment, archived by Brett Wilson of SPARTA in early '06, tested by Ted Faber at ISI.**
 - Bit rot – the testbed evolves, do the experiments in the archive?
- **Complete complex experiments, for serious users**
 - Purpose 1: Advance science by supporting **repeated research**.
 - Purpose 2: Help experimenters: **morph** an existing experiment to meet new objectives.

3. Security Experimenter's Workbench





Controls

File View

FloodWatch/EMIST2

- Traffic Generation
 - Replay
 - Harpoon
 - gen1
 - DNS Generator
 - ICMP Generator
- Attack Agent
 - Flooder
 - Cleo
- Defense
 - FloodWatch
 - Cossack
 - DWard
- Data Processing
 - TCPDump
 - Connection Analysis

Participating Nodes and Settings

Nodes: gen0A gen0B gen1A gen1B

Server Sessions: 30

Client Sessions: 3

Warp Factor: 0.00

Server Port: 9090

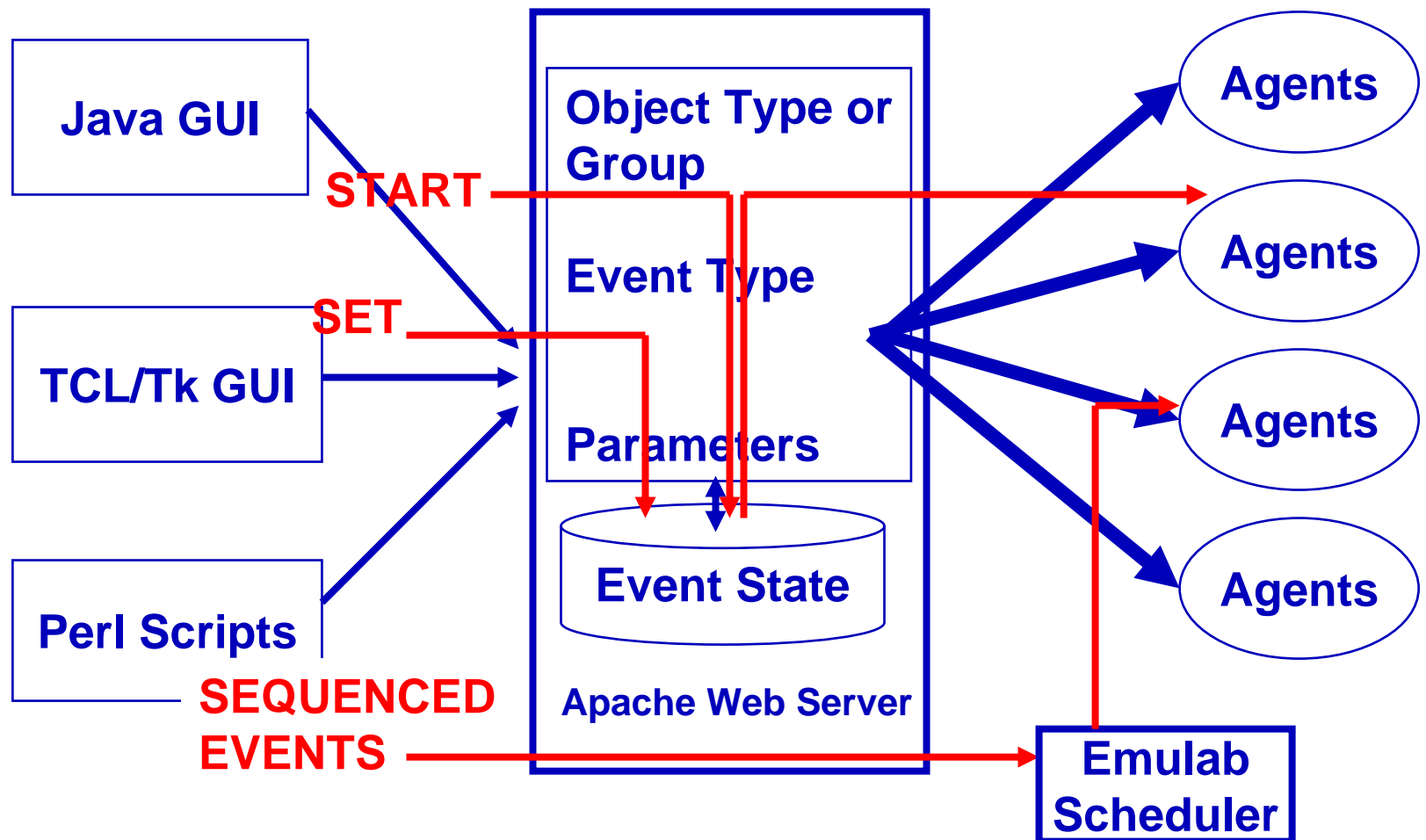
Node Networks

Node	Network
gen0A	11.0.0.0/24
gen0B	12.0.0.0/24
gen1A	13.0.0.0/24
gen1B	14.0.0.0/24

Set Start Stop

```
xmlexec getnodes value: {gen0B=[10.1.2.4], gen0A=[10.1.2.3], gen1B=[10.1.3.4], gen1
xmlexec getstate output: ok
xmlexec getstate value: {HARPOON.gen1.gen0B.cidr=12.0.0.0/24, HARPOON.gen1.ssession
```

Workbench Internals





Workbench Feature Backlog

- **Integrate Instrumentation such as packet counters into displays**
 - Requires push vs. pull or telemetry stream model
- **Integrate Topology information into displays**
 - Direct input based on selection from graphical topology
- **Analysis Tools**
 - Integrate simultaneous and post-mortem processing
 - » Formats: tcpdump, netflow, etc.
 - Long term data capture
 - » Migrate data/packet capture from local experiment disks to users:/proj or other persistent filesystem
 - » Large volumes of data require data management
 - Integrated display of analysis tools output



Workbench / Automation Requirements Gathering

- **DDoS requirements generalize to broader Security Experiment Requirements**
 - Need systematic examination of Worm, Routing, etc.
- **Automate whenever possible for:**
 - Repeatability, efficiency, ease-of-use
- **Experiments described in Emulab ns2 format may include primitive events**
 - Under base Emulab system, provides control for a very limited number of operations
 - Need support for extensible (user experimenter) defined events
 - ns2 extensions to specify agents (traffic, instrumentation) in syntax that fits expectations of ns2 user community
 - Future migration to XML representation



Hardware Appliances and Routers

- **DETER requirement: support the experimental test and evaluation of appliances**
 - Commercial products often packaged as appliances, critical future user segment
- **EMIST requirement: high-speed appliances stress the testbed and the tools supporting our methodology**
 - Requirements:
 - Provide the ability to seamlessly integrate appliances as nodes in testbed experiments
 - Stresses all aspects of our methodology at line-rate
 - » Topology – Gigabit forwarding routers (Juniper)
 - » Traffic – Greater aggregation
 - » Data Capture – vanilla TCPDUMP inadequate



DETER Hardware and Appliances Process

- **Develop a systematic process for integrating hardware devices and appliances in the DETER testbed and within EMIST experiments:**
 - Hardware connection
 - Control plane ns topology
 - Control plane manual configuration
 - Data plane manual configuration
 - Control and Data plane semi-automatic configuration (scripting)
 - Control and Data plane automation
 - » Integrate generalized scripts behind the scenes into DETER and EMIST tools
- **Workbench needs extensible support to enable users to add custom support for their devices**



Juniper Routers

- **Deter has 5 Juniper M7i routers**
 - 4 Gigabit Ports/Router
- **The Juniper routers are first-class DETER experimental devices***
 - Can be allocated into an experiment by *Assign*
 - Experimenter's gain access to Junipers
 - Can be assigned IP addresses within ns topology
 - Scripts configure router to use the assigned IP addresses/subnets
 - Static routes configured
- **Requirement: workbench support for routing protocol selection, configuration, change**



CloudShield Appliance

- **A CloudShield Appliance with 4 Gigabit interfaces has been added to DETER as an experimental device**
 - Can be allocated into an experiment by *Assign*
 - Must be configured manually
 - Mapping of interfaces into an experiment is difficult since there are no exposed MAC or IP addresses
 - Usage is complicated by the transparent bridging function that causes the DETER switches to go into layer 2 loops.
 - Spanning Tree Protocol (STP) is disabled on DETER
- **Several hurdles to overcome before we can safely and seamlessly add appliances to the testbed and support them within workbench**

CloudShield CA2200



- **2 Intel IXP2800 network processors running CPOS**
 - 16 microengines each
 - 1 StrongARM processor each
- **Dual Pentium-III management processor running Red Hat Linux**
- **4 GigE or 4 Copper 10/100/1000 network interfaces**



Workbench / Repository Integration

- **Three step experimentation process:**
 - (1) define
 - (2) execute
 - (3) archive
- **GUI/scripting control via agents and events is the middle part of the security experimentation process**
- **Definition of experiments: by hand, or via *ad hoc* scripts to construct ns files**
 - Should we pursue a make-like configuration facility for experiments?
- **Archive**
 - Experiments in the archive need to be more than ns file and the source or binary code
 - What meta-data belongs in the archive?
- **How do we exploit the archive meta-data? Do we directly integrate it into our “make experiment configuration” tool?**



Questions

- **What do we need to make the testbed more useful?**
 - **1. For security research**
 - **2. To reduce the learning curve**
 - **3. To make experienced users more productive**