Sandia SRS Red Team Results

Information Design Assurance Red Team

John Clem

Kandy Phan

DARPA SRS PI Meeting 15 Dec. 2005

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Outline

- IDART™ Objectives
- Initial Analysis
- Results PMOP, CORTEX, PASIS
- General Observations
- Lessons Learned
- Q&A

IDART Objectives

- System Analysis
  - Increase system understanding
  - Test system responses to adversarial inputs
  - Attack assumptions/claims
  - Confirm strengths and reveal weaknesses
- Red Team
  - Open…
  - Flexible
  - Objective
  - Fair

Initial Analysis

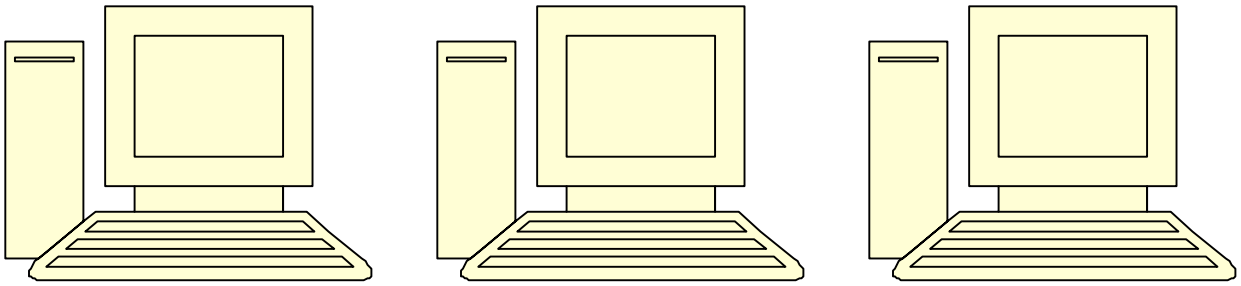
- Reviewed three SRS technologies for live red team readiness
- Two technology development projects were chosen for a live red team engagement
- One technology project was chosen for an attack brainstorm only
- Criteria
  - Technology implemented?
  - Stable?
  - Potential for tangible results?

PMOP

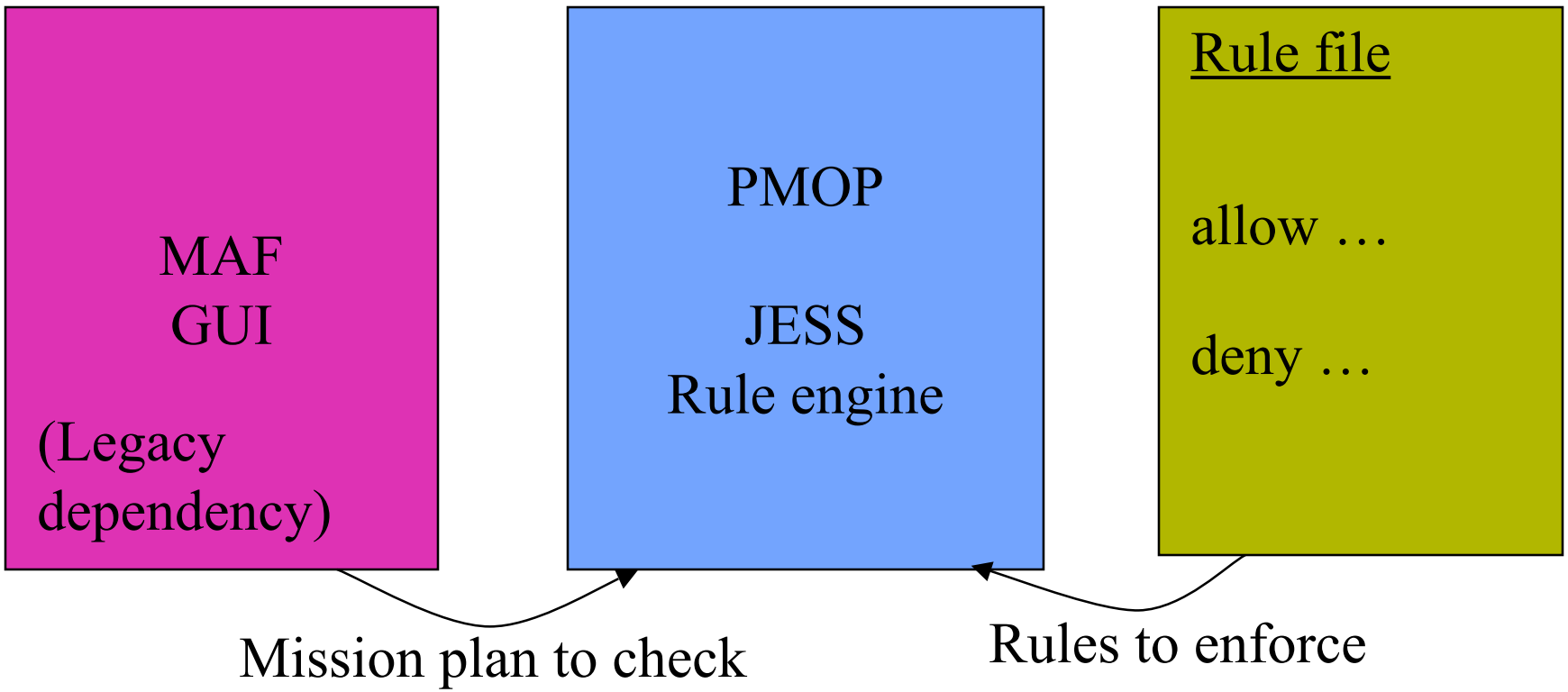
- **Adversary Model:**
  - **A regular user with malicious intent**
  - **Operating system vulnerabilities are out of scope**

PMOP (2)
Targets

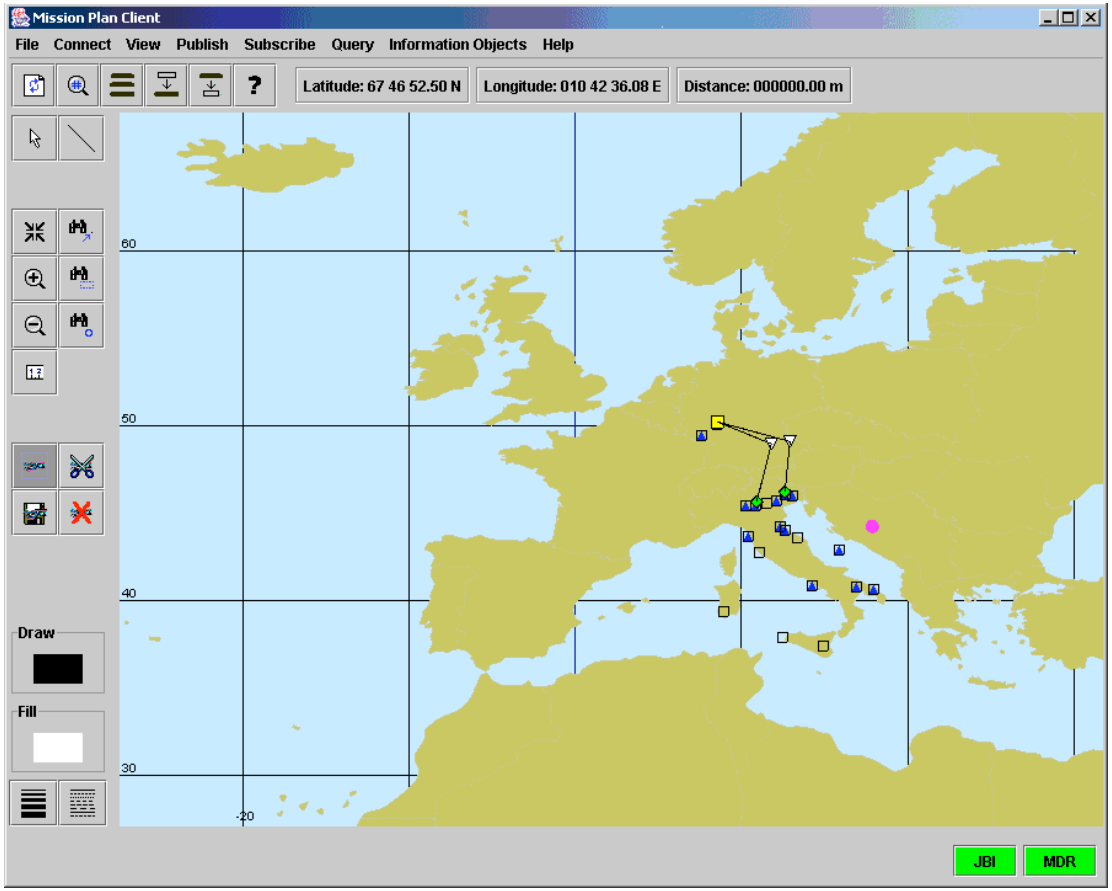
- 3 Separate Components on 3 systems
  - 1. Rule System
  - 2. “File Save As …” Dialog Box
  - 3. Wrapped Shell



PMOP (3)
Rule System



PMOP (4)
MAF GUI Client



PMOP (5)
Example Rule File

```
(defrule MAIN:first-leg-must-be-a-takeoff
        (MISSION_EVENT_ROW (EVENT_TYPE ?&~"TO")
        (EVENT_SEQ_ID ?id1) (prev -1))
        =>
        (error-feedback "first-leg-must-be-a-takeoff " ?id1))
```

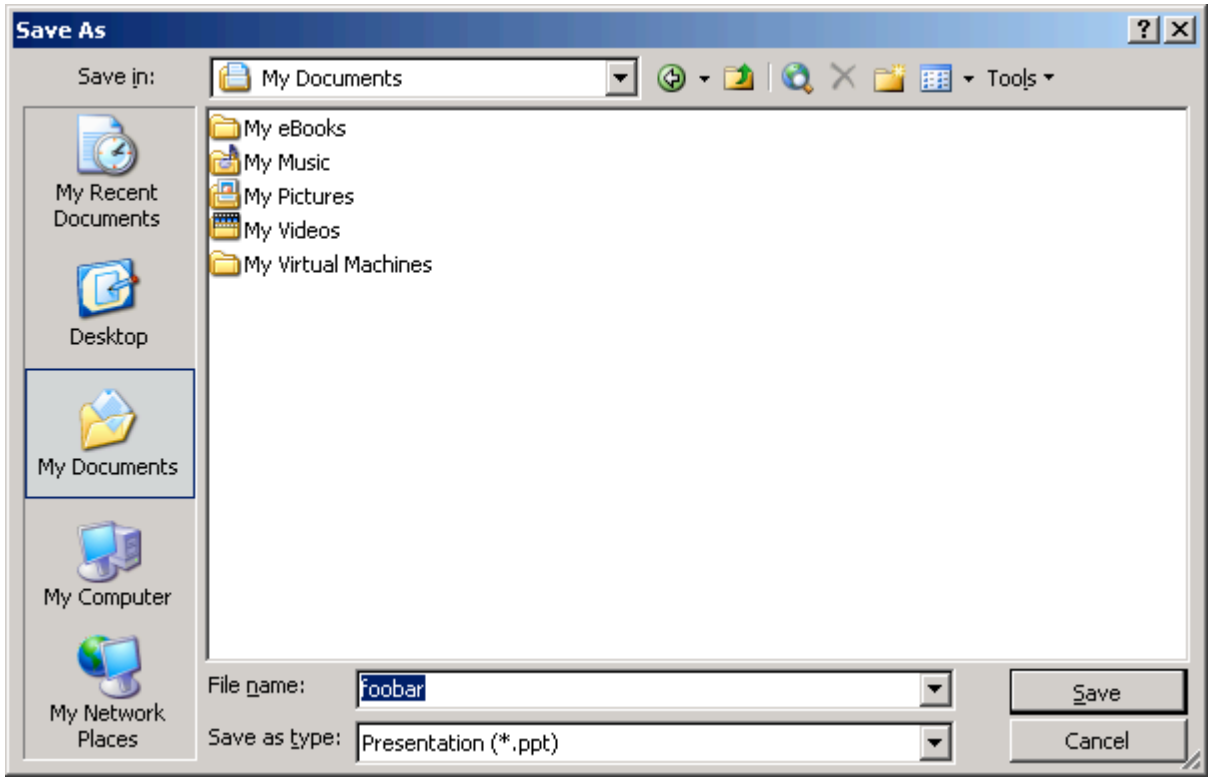
```
(defrule MAIN:aircraft-cannot-exceed-supported-weight-of-airbase
        …
         (WEIGHT ?acweight&:(> ?acweight ?abweight)) )
        =>
        (error-feedback "aircraft-cannot-exceed-supported-weight-of-
        airbase "))
```

PMOP (6)
Rule System

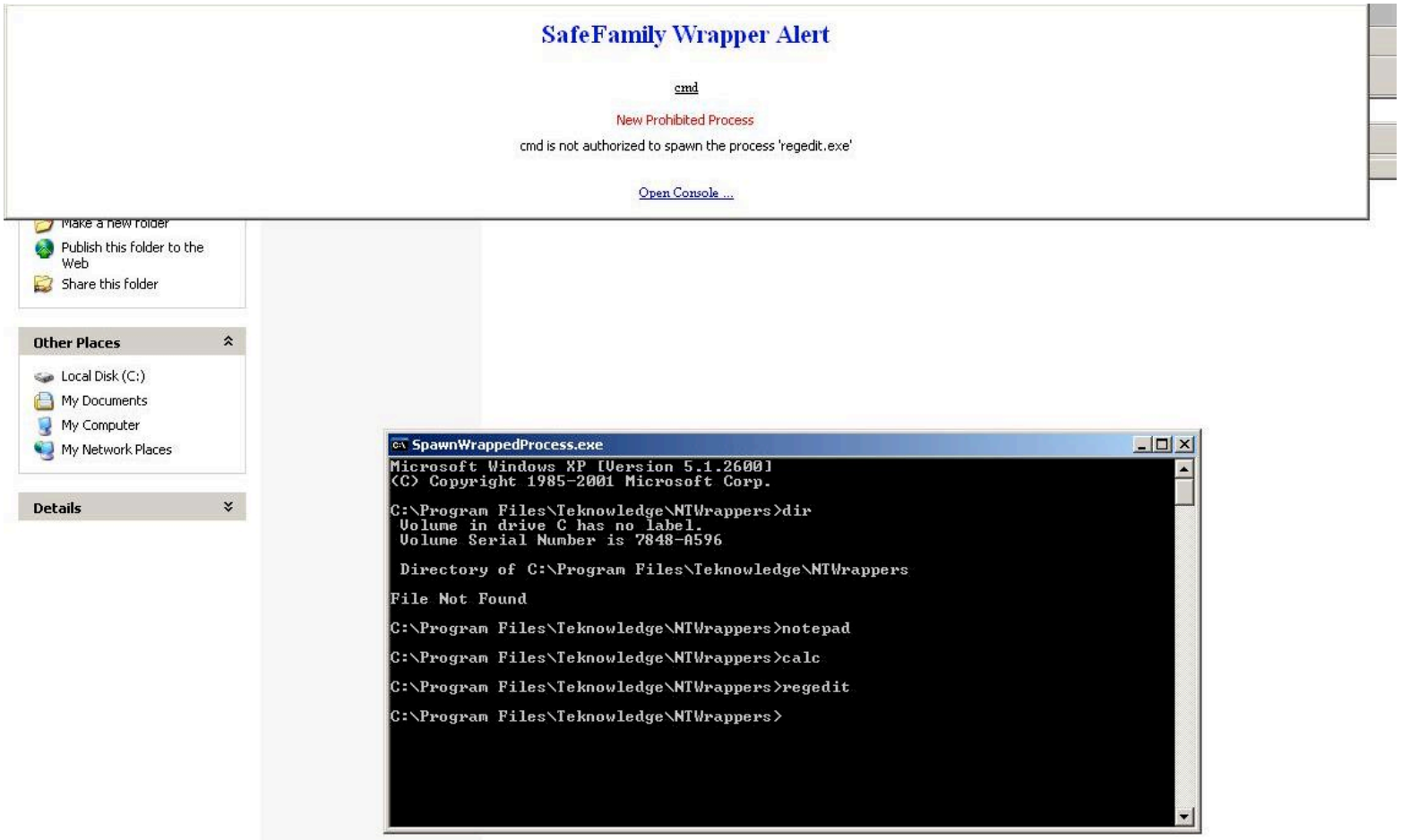
- **Strengths:**
  - **Fast**
  - **Accurate**
  - **XML**

- **Weaknesses:**
  - **Need stronger input validation (e.g. XML)**
  - **Scalability/Consistency of rules**
  - **Domain/Expert knowledge dependent**

PMOP (7)
"SaveAs" Dialog Box



PMOP (8)
Wrapped Shell



PMOP (9)
Wrapper Config File

```
authorize connect  in ws2_32.dll   with Inst_connect
authorize bind     in ws2_32.dll   with Inst_bind
authorize sendto   in ws2_32.dll   with Inst_sendto
authorize recvfrom in ws2_32.dll   with Inst_recvfrom

// mediators for MSO SaveAs and Open Dialogs
transform FindFirstFileExW   in kernel32.dll with Inst_FindFirstFileExW
transform FindNextFileW      in kernel32.dll with Inst_FindNextFileW
monitor   FindClose          in kernel32.dll with Inst_FindClose
```

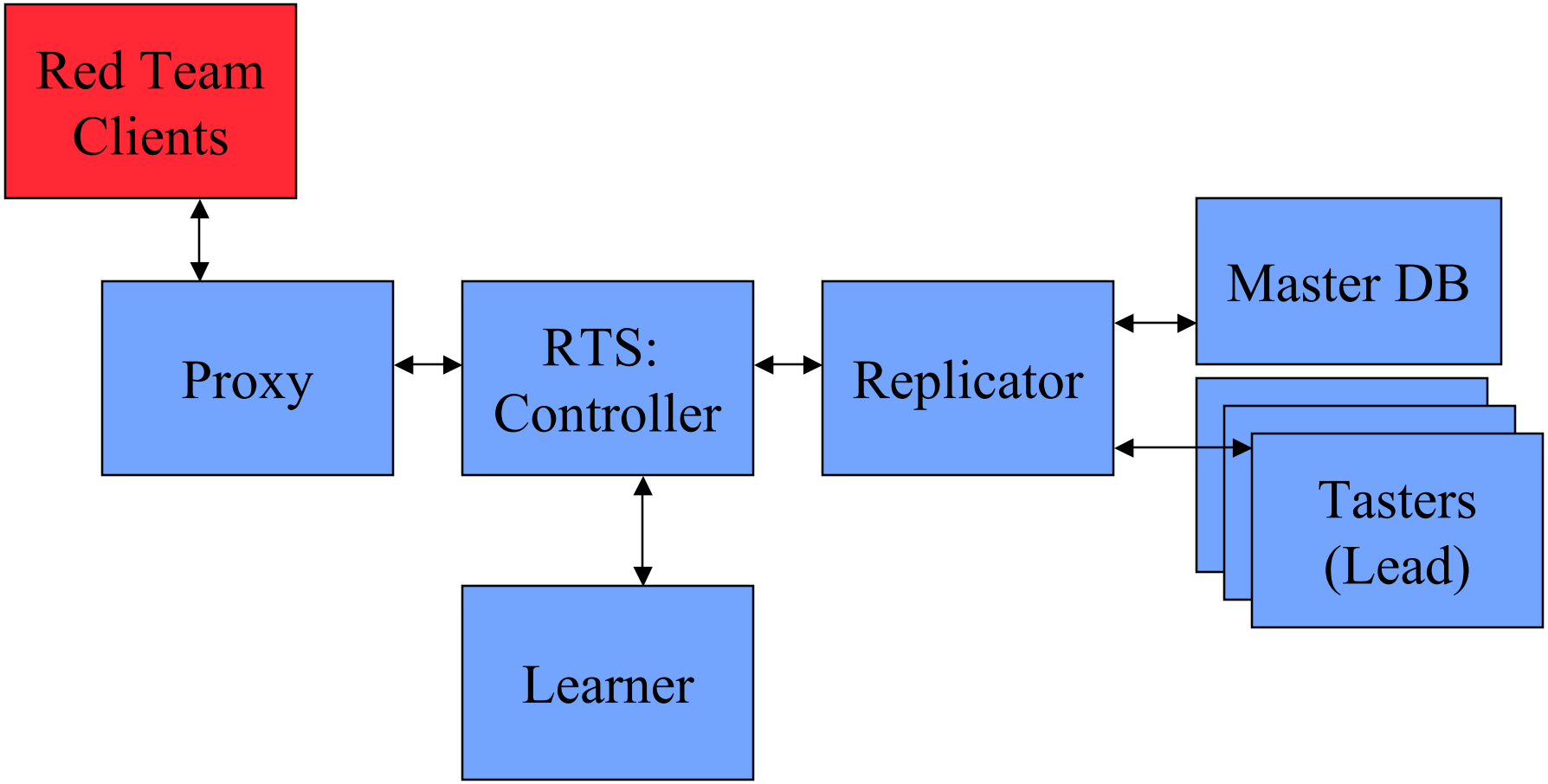
PMOP (10)
Wrapper Config File

```
<file inherit="true" override="false"
resource="%appdata%\Mozilla\Firefox\profiles.ini">
    <read    action="allow" audit="false"/>
    <write   action="allow" audit="false"/>
    <execute action="deny"  audit="true"/>
    <com     action="deny"  audit="true"/>
  </file>
```

PMOP (11)
Wrapped Shell

- **Strengths:**
  - **Canonicalization of file names**
  - **Granularity**
- **Weakness:**
  - **Scalability of configurations**
- **Results:**
  - **NT wrappers did well protecting the JBI directory**

CORTEX


Red Team Clients
Proxy
RTS: Controller
Replicator
Master DB
Tasters (Lead)
Learner

CORTEX (2)

- **Strengths:**
  - **Fast response/Efficient of learner**
  - **Block mechanism/"Binary poison"**
  - **Scalability in number of tasters**
  - **Single entry point**
  - **Real automatic system**

CORTEX (3)

- **Weaknesses:**
  - **Instrumentation capabilities**
  - **Instability of proxy and controller (buffers?)**
  - **Algorithm to switch tasters**
  - **Invalid error messages**
  - **Failure detection for tasters**

CORTEX (4)

- **Red Team flags:**
  - **1. Crash system twice with same attack**
  - **2. False positives**
  - **3. Take down system**

- **Results:**
  - **Flag 1 not achieved**
  - **Flag 2 achieved**
  - **Flag 3 achieved**

- **Instability did not allow full testing or attribution of effects**

PASIS

Increasing Intrusion Tolerance via Scalable Redundancy

General Observations

- **Strengths**
  - **Provable guarantees assuming limited number of Byzantine servers and *unlimited* number of Byzantine clients**
  - **Invisible to ordinary user**
  - **Very efficient in normal operation**
  - **Plausible attack requires sophisticated adversary with extensive real-time knowledge of network state**
  - **Sensible implementation successfully thwarts obvious lines of attack (timestamp manipulation, message replays)**

PASIS (2)

- **Weaknesses**
  - **Presupposes extensive PKI**
  - **Interactions with underlying file system implementation can be complex, hard to specify, could undermine liveness/linearizability guarantees**
  - **Possibility of large overhead, adversary can force system to do a lot of redundant work (live engagement needed to confirm this)**
  - **Not entirely clear how to update system while running (add/drop servers, change parameters or algorithms)**

PASIS (3)

Attack Brainstorm Results (attack graph)

PASIS (4)

Conclusions

- In theory there may be conditions showing PASIS protocol is not bullet-proof
- Weaknesses are in underlying assumption of scaled PKI; always correct file system interactions; lack of defined maintenance procedures
- Strengths are in strong proofs; transparency; efficiency; significant adversary attack requirements

General Observations
DARPA/SRS

- Red Teams successful with causing false positives
  - Low cost attack
  - DoS
- System states under attack difficult to know
- Some threat models still being used by developers are weak
- System security not inherent
  - Dependent on other things (implementation)
  - What happens when you build a system of systems?

General Observations (2)

- Implementations have been shown to include shortcuts bypassing the theoretical model specifications
- Scoring has pros and cons
  - There can be COI
    - Red Teams discouraged from trying novel attacks due to low likelihood of success
    - Red Team could run up the score based on uninteresting variations of successful attacks
      - Mitigation: White Team oversight

Lessons Learned

- Murphy was here (again)
- Live Red Team experiments/exercises are not low overhead
  - Certain amount of overhead for even one day
- Need stable implementations
  - Homogeneous platforms increases reliability
    - Hardware
    - OS
    - Applications
  - Redundant platforms improves efficiency
  - Certain metrics difficult to measure without this

Lessons Learned (2)

- Frozen version
- Need developer instrumentation to understand system states during red teaming
- Advantageous for developers to have/consider more sophisticated threat models early
- Red Teams need/use shortcuts to adequately model adversary pressure
  - These are exercise assumptions
  - This adds value/reduces cost

Value Added

- Different Perspective (Malicious)
- Experience
- Clarifies understanding
- Provides new insights
- Structure for analysis

Q&A/Discussion

IDART™ Contact Information

John Clem
jfclem@sandia.gov
505-844-9016

Kandy Phan
kphan@sandia.gov
505-284-6802