

# Impediments to Building Survivable Systems: An Experience Report

Paul Rubel



# Survivability is Difficult

- If you react to attacks you're continuously playing catch-up.
  - There are too many attacks to deal with one by one.
- A better approach is to deal with the symptoms or effects of attacks.
  - Bandwidth consumption
  - File modification
  - Resource abuse

# Creating Survivable Systems

- We need to look beyond functional requirements for applications.
  - Survivability needs to be part of the design.
- It can be hard to validate the survivability of a system.



# Adding Survivability Using Middleware

- QuO is a middleware system that offers an application the ability to adapt to the changing environment in which it is running.
  - Host resources (CPU and memory) usage
  - Network resource availability
  - Intrusion status
- It can be added into a distributed object application with minimal impact on the application.

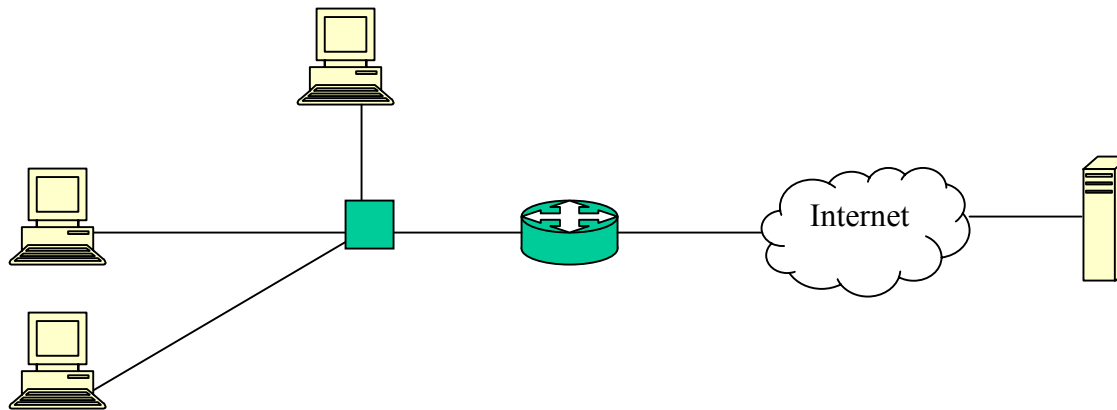
# Applications that Participate in their Own Defense (APOD)

- By adapting to and trying to control its environment, an application can increase its chances of survival under attack
  - We use QuO to integrate multiple defense mechanisms into a coherent strategy for adaptation and defense.
  - We use mechanisms, where they exist, to harden or protect an application, a resource, or a service.
- Using middleware allows us to develop functionality and survivability separately.
  - Where interaction is necessary hooks are provided through an explicit interface.

# Defense Mechanisms Used in APOD

- Network and Host Sensors
  - Snort is a lightweight network intrusion detection system
  - Tripwire for detecting file systems integrity violations
- Actuators
  - Network traffic filters - Iptables
  - File systems recovery – secure backup
- Dependability management using replication
  - AQuA – replication system from University of Illinois, Urbana-Champaign
  - APOD Bus – mechanism for publishing data about application's status and for maintaining replicas of application processes
- Bandwidth Management
  - Intserv (RSVP, SecureRSVP) and Diffserv
- Access Control
  - NAI's OO-DTE at the interceptor layer

# Examples of APOD Strategies



- **Containment:** Use snort and iptables to detect and limit the extent of attacks
- **Outrun:** Use dependability with replication to move away from attacks

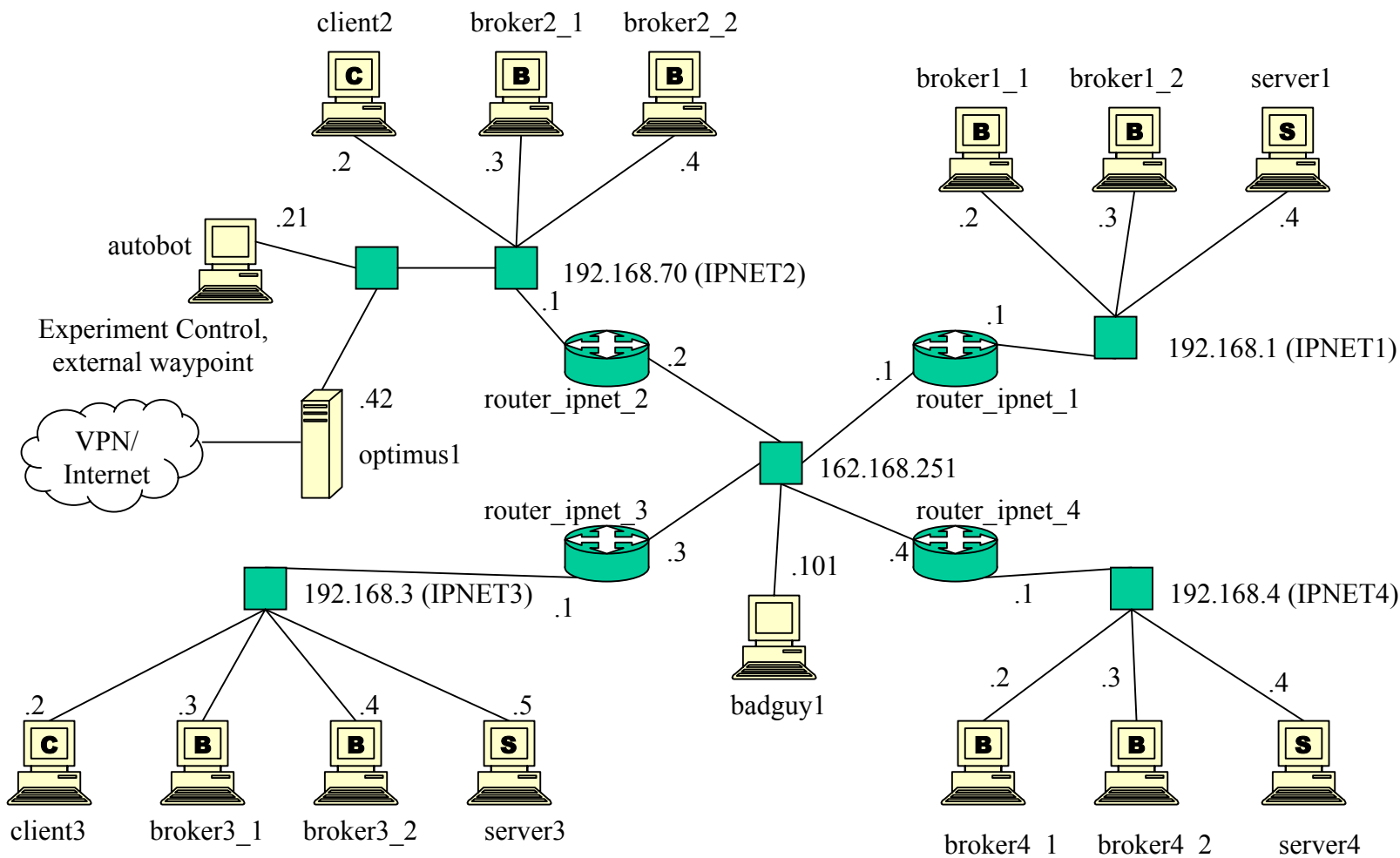
# Red Teaming APOD

- An APOD enabled application is currently being evaluated by a red team from Sandia labs.
- They are testing the added value of APOD to an application.
- We use work factor to test/validate claims.
- Work factor is a good cross cutting measurement.
- How do you compare across teams or even runs?
- A testing plan works but only until you need to test something new that wasn't in the plan of the previous tests.

# The Red Team Target

- The application has three pieces: client, image server, and broker.
  - The broker is responsible for hooking clients requesting images to the server that can provide those images
- Two broker components are maintained by the APOD bus on a set hosts.
  - We using snort and tripwire as sensors and iptables and the bus to react to attacks.

# Experiment Topology



# Survivability Has Open Issues

- Conceptual Issues
  - How do we define or validate a survivable system?
  - How do we compare or evaluate a survivable system?
- Practical Issues
  - Interoperability between survivability mechanisms
  - Reuse of components

# How Do We Define and Validate Survivability?

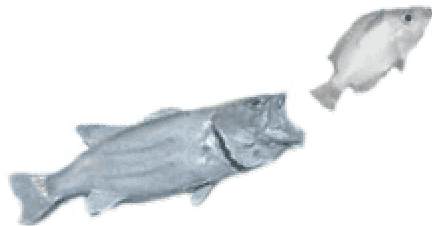
- Security and survivability are not taken as seriously as functional requirements.
- Survivability is much harder to test for. How do you know if you have met your goals in terms of coverage, liveness, and have no competing goals?
  - Members of the ITUA (Intrusion Tolerance Using unpredictable Adaptation) project are currently looking into validating survivability as part of our project.
- Survivability and security cross cut many issues.

# Practical Impediments Need To Be Overcome

- How do you keep mechanisms from being used against each other?
- Need to be able to integrate all your mechanisms together and have them work with each other.
- Issues that effect integration:
  - ability to resolve conflicts
  - compatibility of interfaces
  - compatibility of data formats
  - ability to integrate separate software
  - usability of the components

# Adaptation is Necessary for Survival

- To survive, the system must be able to react to changes and cope with them using situational awareness.
- This isn't simple, especially in distributed systems.
- We need to decide what to listen to and then how much trust to place in what is said.
- Predictable reactions can be exploited by an attacker. The ITUA project is working on unpredictable adaptation.
- However unpredictability makes measuring more difficult.
- How much can one adapt, what is the trade-off between a good choice and unpredictably choosing a non-optimal one.



# Current Status

- APOD is currently undergoing red team testing.
- With more applications and experience we hope to distill commonality out and make the defense enabling process easier.
- New mechanisms are being developed as we are exposed to new kinds of attacks.

# Conclusion

- We have made good progress in making a survivability toolkit: [apod.bbn.com](http://apod.bbn.com).
- Validation using red teams is a good process but finer grained measurements would be useful.
- A middleware approach to survivability allows separation of concerns.
  - We still need to add specific domain knowledge to each application though.

# Thanks

- This presentation and the work it describes would not be possible without the hard work of: Partha Pal, John Zinky, Chris Jones, Franklin Webber, Michael Atighetchi, Joe Loyall, Rick Schantz, and many others.
- Contact Information:
  - Paul Rubel - [prubel@bbn.com](mailto:prubel@bbn.com)
  - Project Web Sites
    - [apod.bbn.com](http://apod.bbn.com)
    - [itua.bbn.com](http://itua.bbn.com)
    - [quo.bbn.com](http://quo.bbn.com)