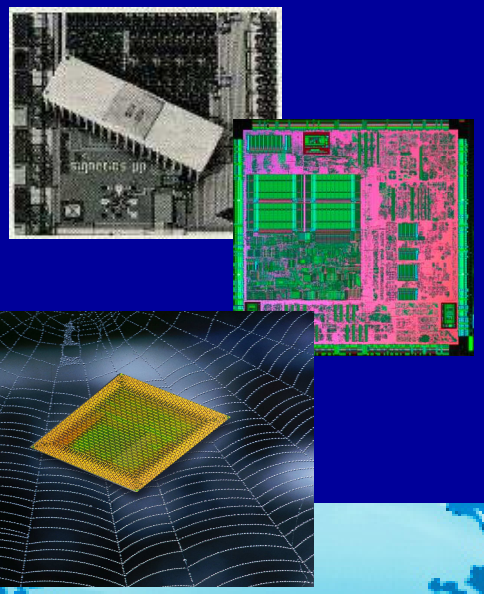
School of Electrical and Computer Engineering
Georgia Institute of Technology

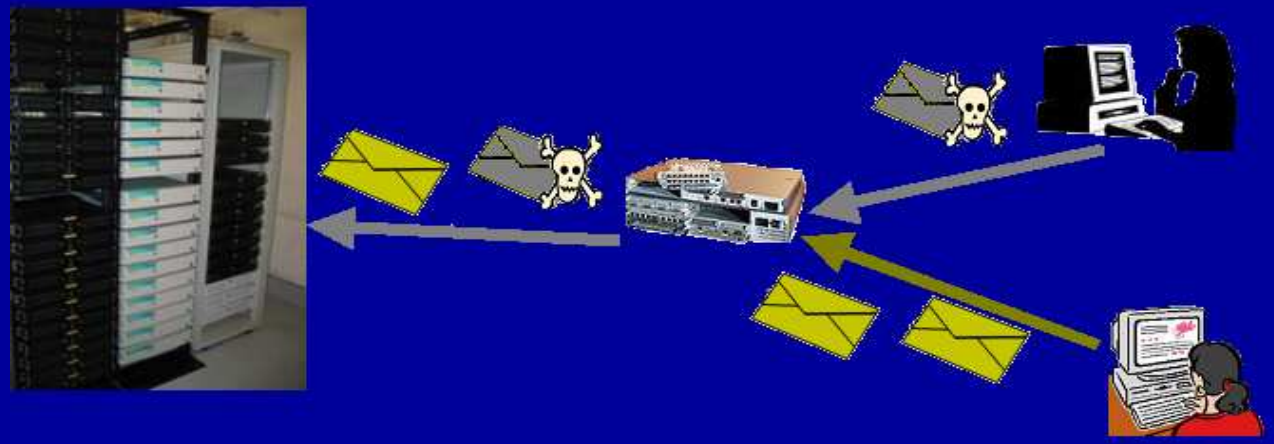
An Intrusion-Tolerant and Self-Recoverable Network Service System Using A Security Enhanced Chip Multiprocessor

Weidong Shi (GaTech)
Hsien-Hsin (Sean) Lee (GaTech)
Guofei Gu (GaTech)
Laura Falk (Michigan)
Trevor N. Mudge (Michigan)
Mrinmoy Ghosh (GaTech)



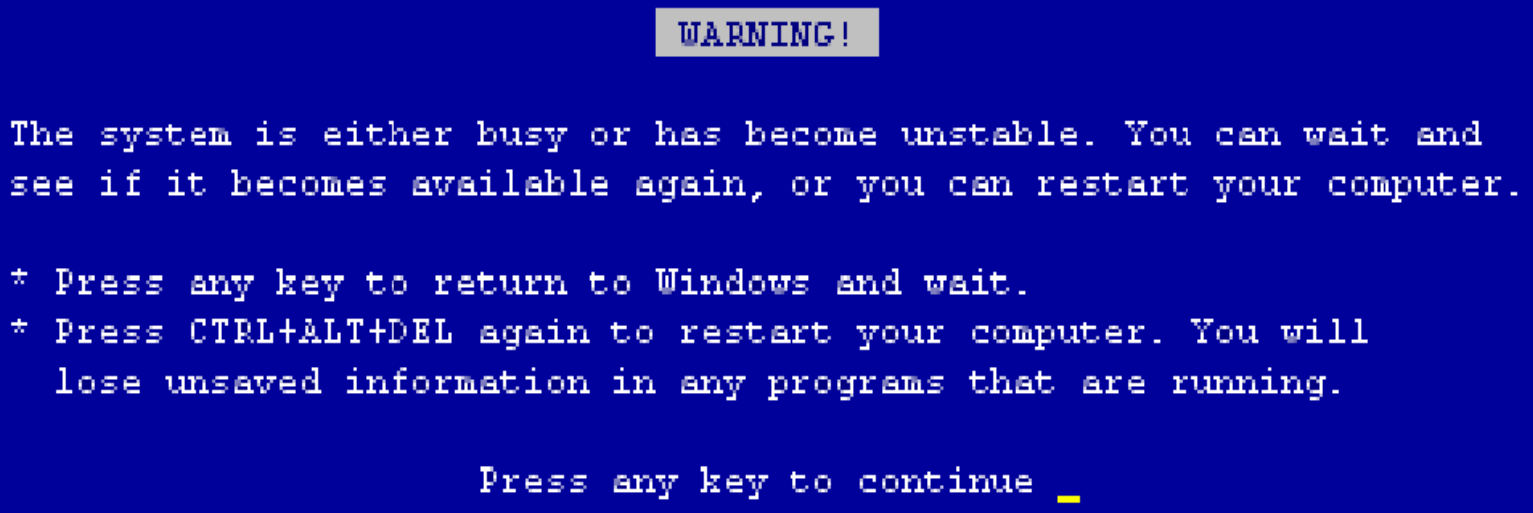
Threats to Network Services

School of Electrical and Computer Engineering
Georgia Institute of Technology



- Server receives (multiple) flows of requests from both malicious and legitimate users
  - Crash-based Denial-of-Service (DoS) attacks
  - Buffer overflow attacks

Traditional Solutions for Lost Services



- Take service off-line and wait for patches
- Termination-reboot
  - Expensive and slow
  - Cant handle DoS exploits (continue crash)
  - Loss of user data
- Intrusion detection and Firewall
  - Identify the source of an attack is non-trivial (IP spoof)

Objectives

- High Availability, Reliability, and Survivability.
- Explore new programming and usage model of the emerging Multi-core processor or **Chip Multiprocessor (CMP)**
- Provide "architectural support" for network services to be
  - Autonomic
  - Remote-attack survivable
  - Self-recoverable
- High Performance

School of Electrical and Computer Engineering
Georgia Institute of Technology

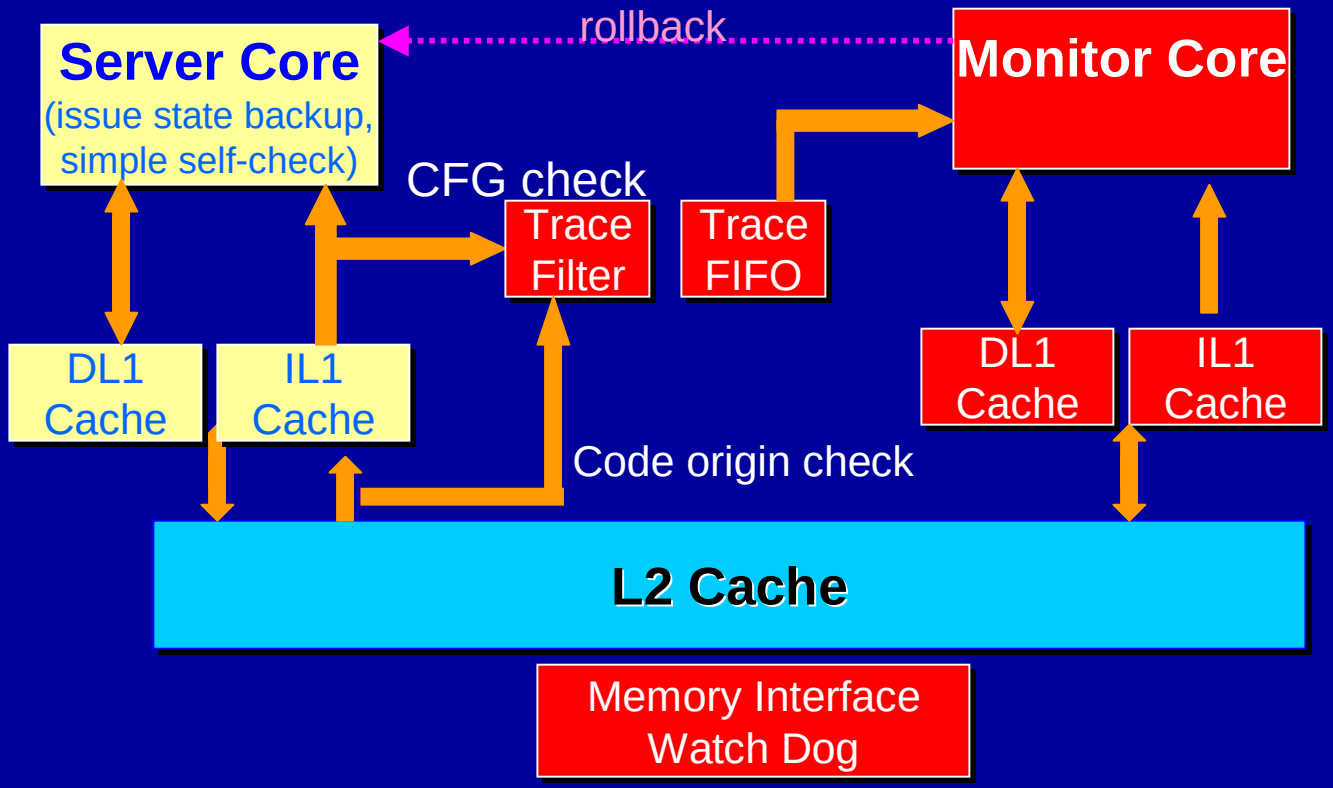
Why Chip Multiprocessor?

IBM POWER5

- Everyone is making it. Why?
- Insulation: Each core of a CMP can be programmed to run at different privilege levels with different OSes.
- Integrated fine-grained processor state monitoring.
- Concurrent monitoring and efficient state backup and recovery.
- Massive multi-core will have many idle cores.

Asymmetric Security Enhanced CMP

School of Electrical and Computer Engineering
Georgia Institute of Technology

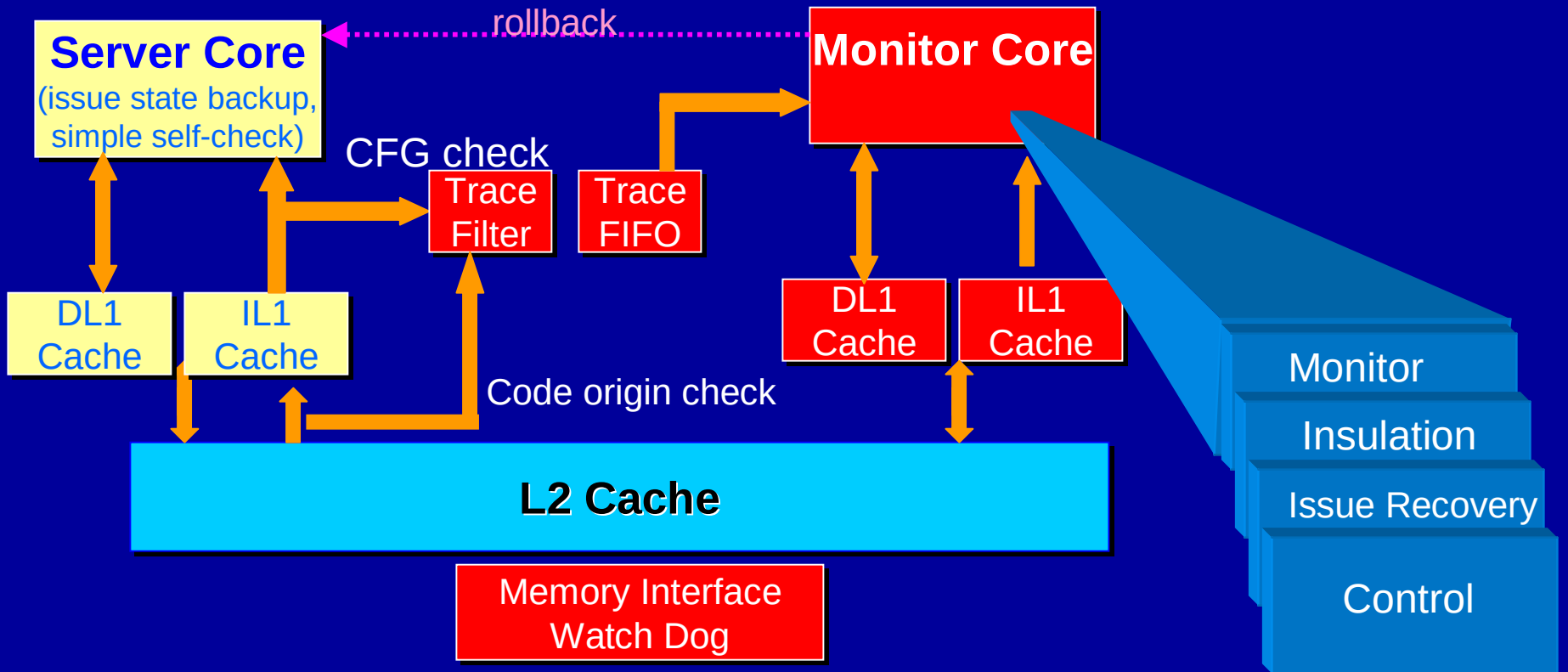


Physical Memory Space (used by service OS and applications)	Protected Memory Space (monitor BIOS, OS, and SW)

Cores Are Managed by Privilege Levels. No SW bypass (hardware insulation)

Asymmetric Security Enhanced CMP

School of Electrical and Computer Engineering
Georgia Institute of Technology



Physical Memory Space (used by service OS and applications)	Protected Memory Space (monitor BIOS, OS, and SW)

Cores Are Managed by Privilege Levels. No SW bypass (hardware insulation)

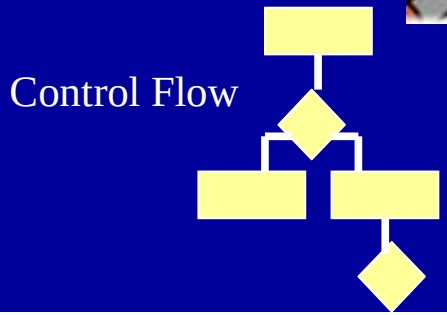
Monitor and Recover Network Applications

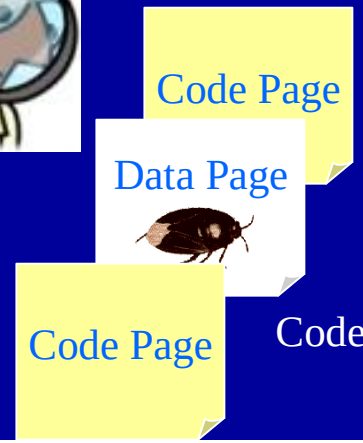
- Network service is request-response oriented
- Monitor core inspects well-being of applications concurrently with application execution
- Rollback application state when corruption/intrusion is discovered
- Continue execution from rollbacked state

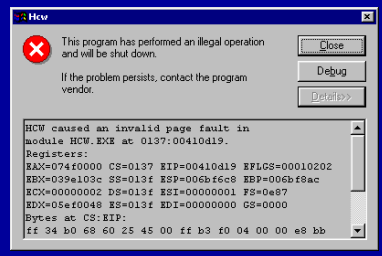
Monitor Core: Inspection

School of Electrical and Computer Engineering
Georgia Institute of Technology

- Verify “Code Origin”
  - only execute code originally loaded from hard drive
  - detect injected code in data pages
- Verify “Control Flow”
  - computed function call matched against valid function entry points
  - detect overflow of function pointers
- Signify “Illegal Operations”
  - monitor handles memory exceptions (bus error, seg fault) first before application OS
  - detect memory corruption



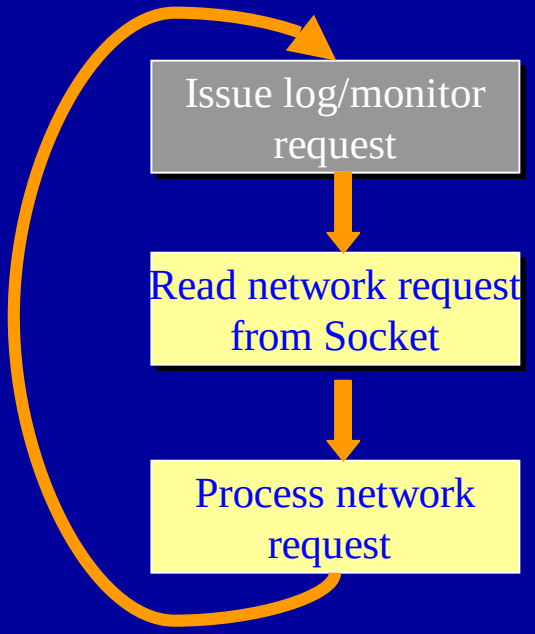




Exceptions

Server Core: Instant Recovery

School of Electrical and Computer Engineering
Georgia Institute of Technology



Abstraction of Server Application

- Maintain memory state, system resources (file handles, locks, semaphore, etc) between processing each network request.
- Upon detection of intrusion/memory corruption/illegal operation, monitor core
  - triggers recovery process
  - rolls back to a known good state (before processing the bad request)
- Continue to handle the next network request.
- Support multiple “backup” states and iterative rollback.

School of Electrical and Computer Engineering
Georgia Institute of Technology

Monitor Core: Incremental Memory Log

- Monitor core maintains separate logs of memory updates triggered by each network request.
- Snoop memory interface
- Can be very fast
  - temporary data on stack does not require backup
  - group memory updates in registers (XMMX) and write them back to RAM directly bypassing caches
- Only backup memory updates to a limit (e.g., a few million most recent updates)
- No source code instrumentation required.

School of Electrical and Computer Engineering
Georgia Institute of Technology

Monitor and Recover Server OS from Rootkits

- How do they work?
  - Patch server OS's interrupt handler table with malicious code pointer
  - Redirect server OS's system call table
- Why are they bad?
  - Hide hacker's traces
  - Give a false well-being image of the system
  - Provide backdoor for the hacker to come back in future
- Rootkits are hard to remove and recover (often need completely reinstallation of the system)

School of Electrical and Computer Engineering
Georgia Institute of Technology

Monitor Core: Backup and Inspection

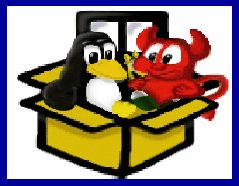
- Backup. Monitor core maintains
  - a clean version of server core's system call table, interrupt handler table in its private space
  - a clean copy of server core's ktext (kernel text)

- Inspection
  - modification to important kernel table structures
  - modification to ktext

School of Electrical and Computer Engineering
Georgia Institute of Technology

Monitor Core: Recovery

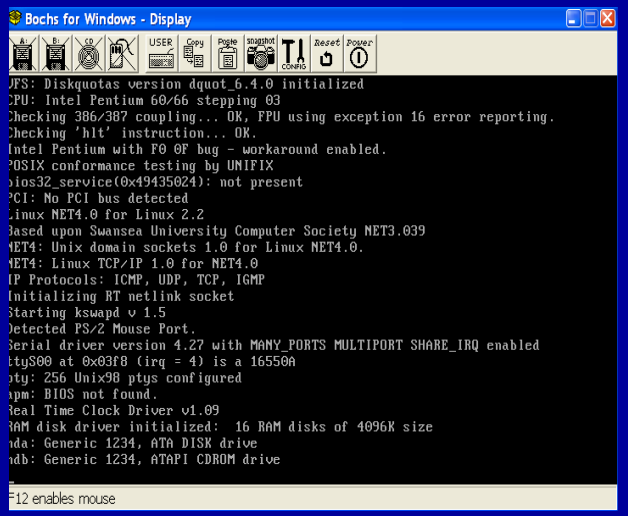
- Patch system call table/interrupt handler table without reboot (use the same technique against the hackers)
- Perform live patch of maliciously altered ktext with the original clean ktext copy
- Must support legitimate system upgrade. Initiate recovery process from a separate management channel by administrators.

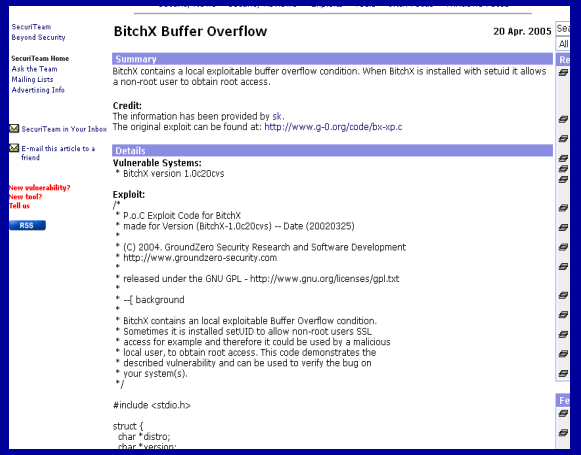


School of Electrical and Computer Engineering
Georgia Institute of Technology

Testbed

- CMP Architecture/System Co-design.
- A x86 system emulation (Bochs) + cycle-based architecture simulator (TAXI)
- Run real OS with real service applications, httpd, ftpd, bind, sendmail, etc.
- Recoverability evaluated by applying real x86 remote exploits from hacker and security websites.





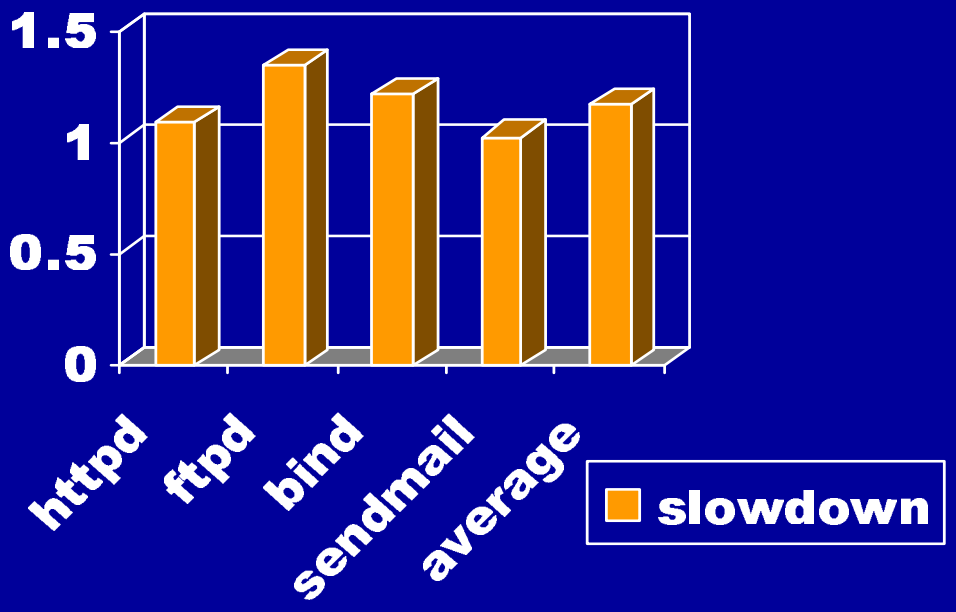
School of Electrical and Computer Engineering
Georgia Institute of Technology

Recovery

- Apply real-world attacks to the emulated server
- Recover from logged states through rollback
- Recovered applications are able to continue responding to new requests
- Related studies done by other schools also show recoverability on per-request basis
- Currently work on robustness evaluation and fault injection

Performance

- Popular server apps. HTTPd, Bind, Sendmail, FTPd
- Tolerable overhead
- 10%-25% slow down of response time



Conclusion

- Combing real-time remote exploit monitor and instant autonomic recovery can enhance service survivability and availability.
- Emerging CMP technique provides redundancy, computing power, and opportunity for new type of autonomic system.
- Non-symmetric CMP with security enhancement can provide improved reliability and availability in the face of remote exploits.
- More research is required to explore the trade-off between availability, performance, architecture design, and cost.

Questions

School of Electrical and Computer Engineering
Georgia Institute of Technology