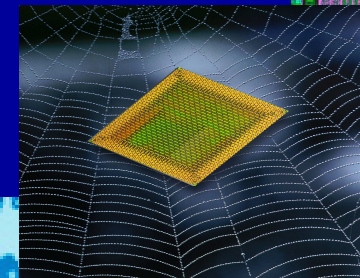
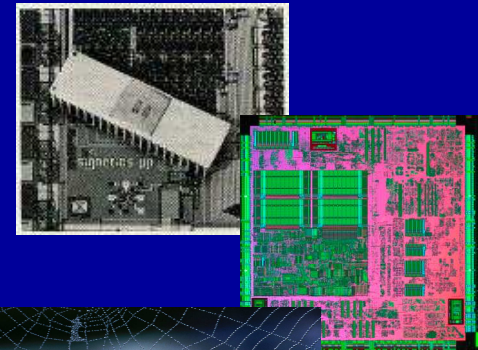


# An Intrusion-Tolerant and Self-Recoverable Network Service System Using A Security Enhanced Chip Multiprocessor

**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



- 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

## WARNING!

The system is either busy or has become unstable. You can wait and see if it becomes available again, or you can restart your computer.

- \* Press any key to return to Windows and wait.
- \* Press CTRL+ALT+DEL again to restart your computer. You will lose unsaved information in any programs that are running.

Press any key to continue \_

- 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



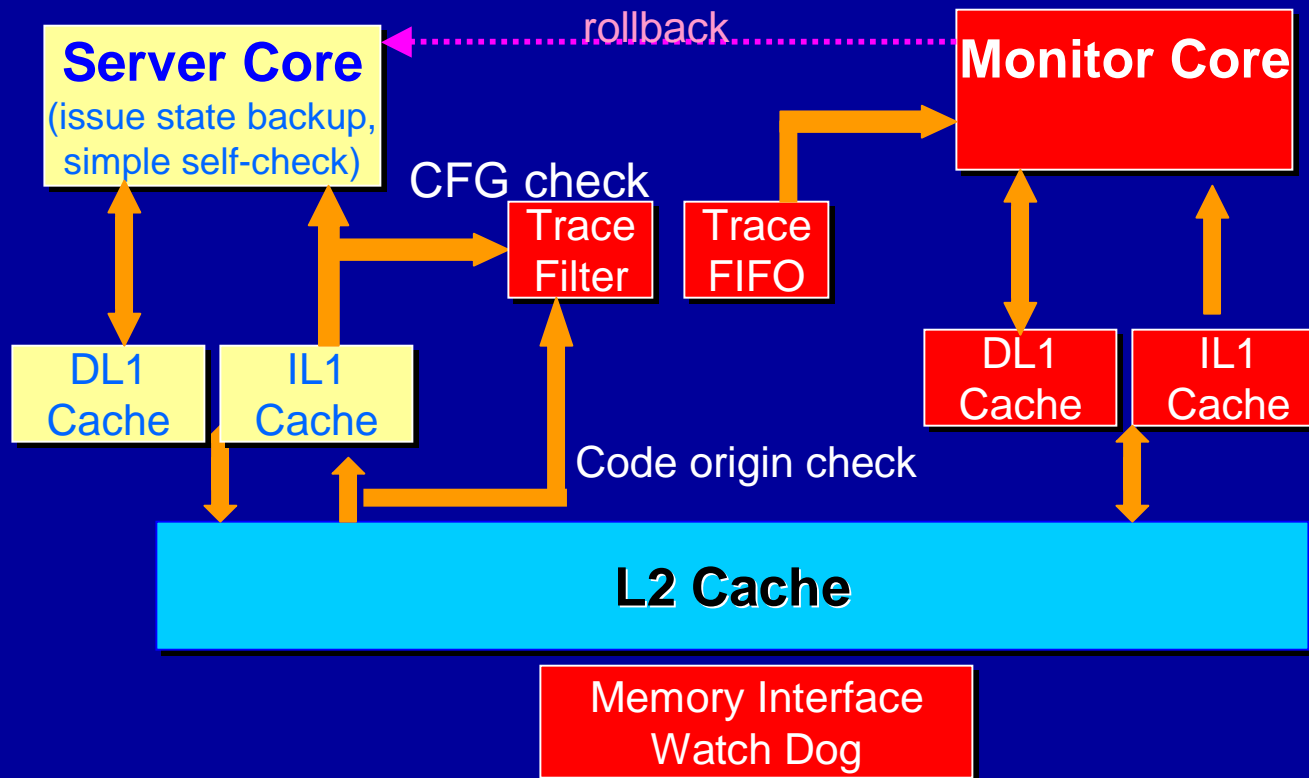
# Why Chip Multiprocessor?

## 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

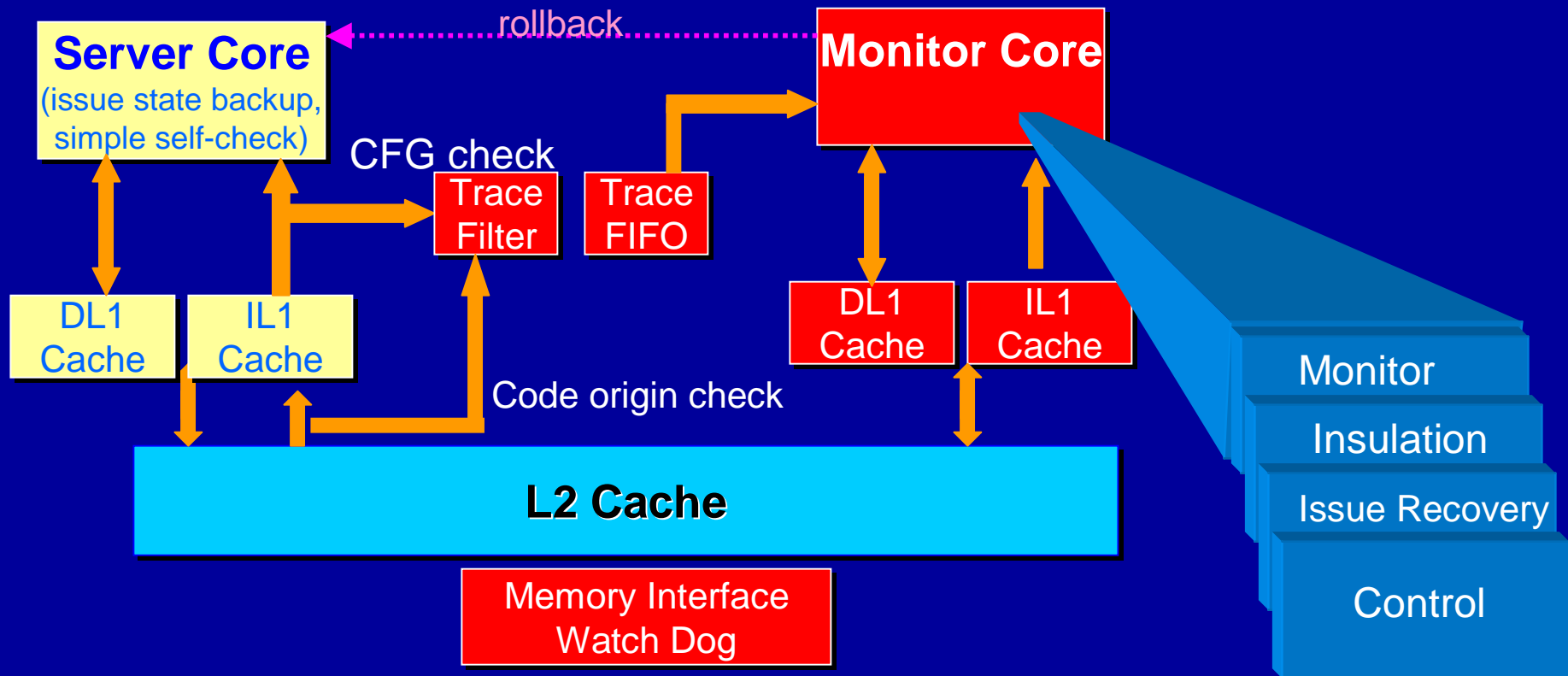


Physical Memory Space  
(used by service OS and applications)

Protected Memory Space  
(monitor BIOS, OS, and SW)

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

# Asymmetric Security Enhanced CMP



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)**

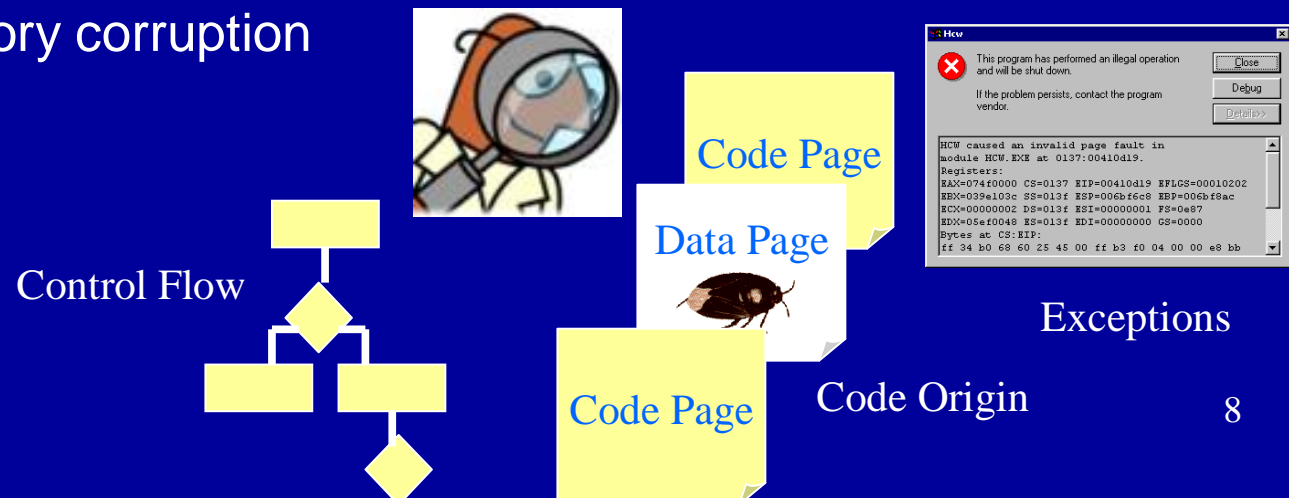
# Monitor and Recover Network Applications

- 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

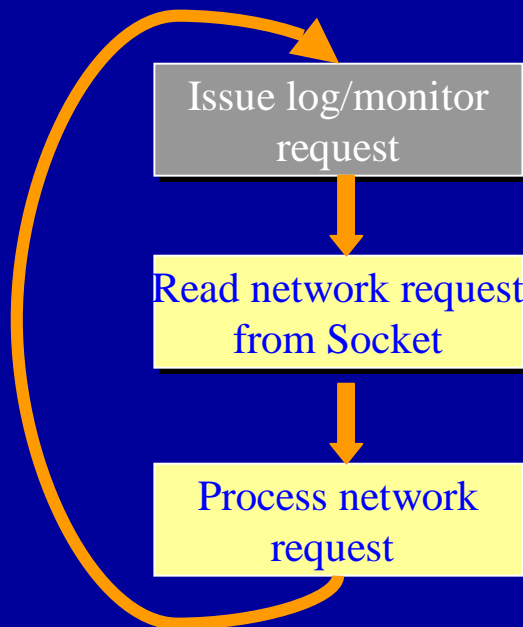


# Monitor Core: Inspection

- 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



# Server Core: Instant Recovery



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.

## 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.

# 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)

# 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

## 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.



# 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.

```
Bochs for Windows - Display
JFS: Diskquotas version dquot 6.4.0 initialized
CPU: Intel Pentium 60/66 stepping 03
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
Intel Pentium with F0 0F bug - workaround enabled.
POSIX conformance testing by UNIFIX
bios32_service(0x49435024): not present
PCI: No PCI bus detected
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
Initializing RT netlink socket
Starting ksquash v 1.5
Detected PS/2 Mouse Port.
Serial driver version 4.27 with MANY_PORTS MULTIPORT SHARE_IRQ enabled
ttyS00 at 0x03F8 (irq = 4) is a 16550A
pty: 256 Unix98 pty's configured
ipm: BIOS not found.
Real Time Clock Driver v1.09
RAM disk driver initialized: 16 RAM disks of 4096K size
ida: Generic 1234, ATA DISK drive
adb: Generic 1234, ATAPI CDROM drive

F12 enables mouse
```

```
SecurTeam Beyond Security 20 Apr. 2005
SecurTeam Home
BitchX Buffer Overflow
Summary
BitchX contains a local exploitable buffer overflow condition. When BitchX is installed with setuid it allows a non-root user to obtain root access.
Credit:
This information has been provided by sk.
The original exploit can be found at: http://www.g-0.org/code/bx-0p.c
Details
Vulnerable Systems:
* BitchX version 1.0c200vs
Exploit:
/*
 * P.o.C Exploit Code for BitchX
 * made for Version (BitchX-1.0c200vs) -- Date (20020325)
 *
 * (C) 2004. GroundZero Security Research and Software Development
 * http://www.groundzero-security.com
 *
 * released under the GNU GPL - http://www.gnu.org/licenses/gpl.txt
 *
 * --[ background
 *
 * BitchX contains a local exploitable Buffer Overflow condition.
 * Sometimes it is installed setuid to allow non-root users' SUDO
 * access for example and therefore it could be used by a malicious
 * local user, to obtain root access. This code demonstrates the
 * described vulnerability and can be used to verify the bug on
 * your system(s).
 */
your-system(s);
}

#include <stdio.h>
struct {
char *distro;
char *version;
}
```

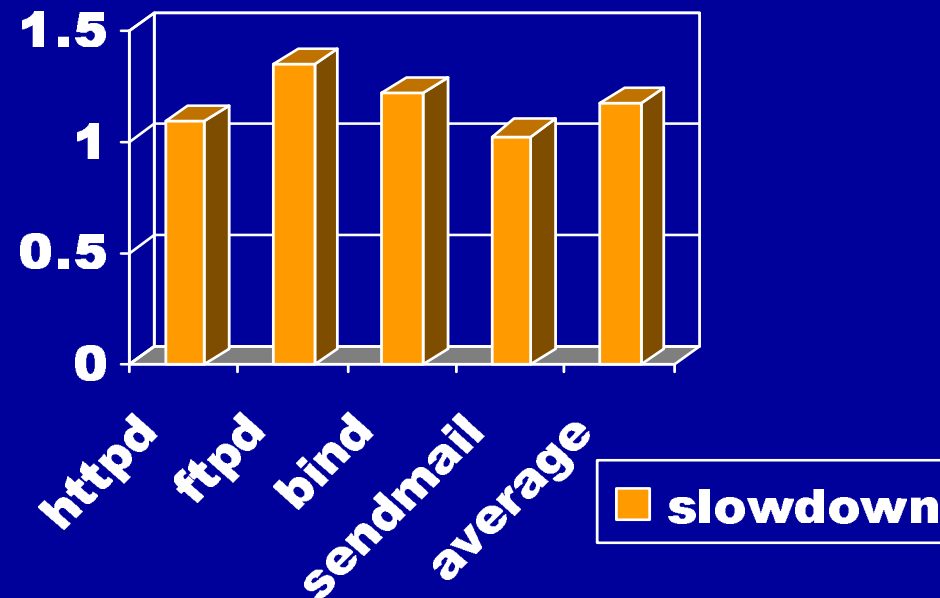
# Recovery

- 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

- Combining 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,<sup>17</sup> and cost.

# Questions

