Connecting the Dot Dots Model Checking Concurrency in Capsicum

ASA-4 21 July 2010

Robert N. M. Watson Jonathan Anderson



Introduction

- UNIX File System (UFS)
- Capsicum: practical capabilities for UNIX
- Whoops, a concurrency vulnerability
- Model checking file system containment
- Conclusions



The UNIX File System (UFS)



The UNIX file system

- Persistent object storage for UNIX
- Hierarchical, user-specified name space
- Also used for IPC
- DAC + MAC
 - A security API



Typical APIs

- Open a file for I/O
 int open(char *path, int flags, ...);
- Change directory

int chdir(char *path);

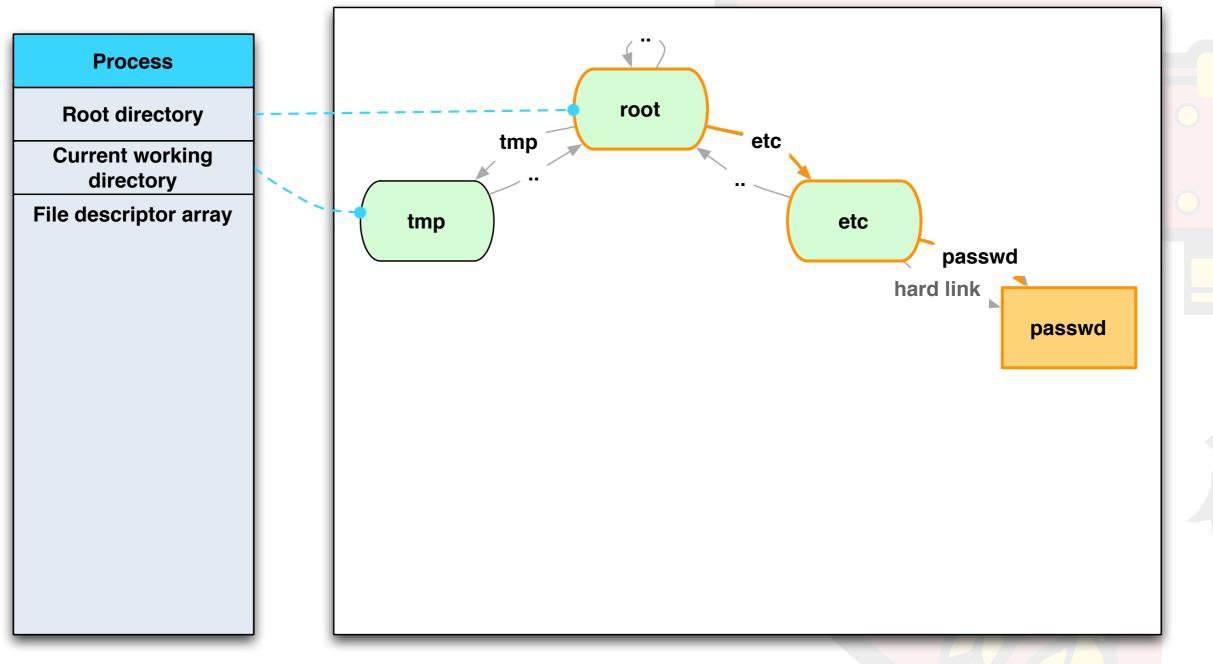
• Rename a file or directory

int rename(char *from, char *to);



Looking up a path

open("/etc/passwd", O_RDONLY)





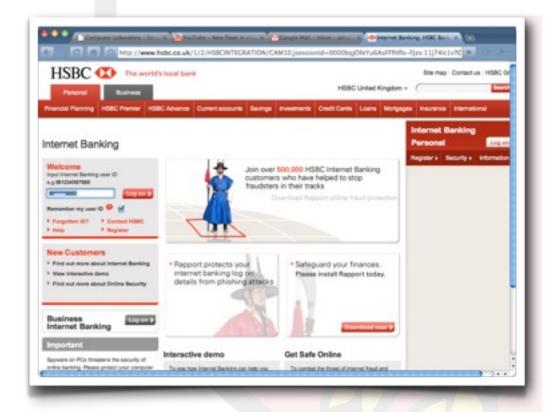
Capsicum: practical capabilities for UNIX



pcoming Deadlines				
DH	Deadlines		Date(s)	Location
Papers	Posters			
2010-02-1		PETS	21 - 23 34 2010	Berlin, OE
3010-02-0		CRIPTO	15 - 19 Aug 2010	Santa Barbara, CA, US
2910-02-2	8	weis	7 - 6 Jun 2010	Cambridge, MA, US
2010-02-2	5	USENIX-LEET	27 Apr 2010	San Jose, CA, US
2010-02-2		wosn	22 Jun 2010	Buston, MA, US
2010-02-2	6	SECSI	26 - 27 Apr 2010	Cologne, DE
2010-03-0	1	CHES	18 - 20 Aug 2010	Santa Barbara, CA, US
2010-03-0	5 2010-05-24	50,95	14 - 16 Jul 2010	Redmond, WA, US
2009-12-1	2010-03-15	USENDI OPTPS	27 Apr 2010	San Jose, CA, US
2010-03-1	9	BCS-HCI	6 - 10 Sep 2010	Dundee, Scatland, UK
2010-03-2	6	CEAS	13 - 14 Jul 2010	Redmond, WA, US
2010-04-0	1	ESORICS	20 - 22 Sep 2010	Athens, GR
2010-04-0	5	ASA	21 34 2010	Edinburgh, Scotland, UK
2010-04-1	6	NSPW	21 - 23 Sep 2010	Cancord, MA, US
2010-04-1	7	ADM-COS	4 - 8 Oct 2010	Chicago, IL, US
2010-00-2	2010-05-18	ACH-SIGCOMM	30 Aug - 3 Sep 2010	New Dehil, 3N
pcoming Conferences				
	Event	Date(s)	Location	URLs
	NOSS	28 Feb - 3 Mar 2010	San Diegs, CA, US	permalinik
	CT-RSA	1 - 5 Mar 2010	San Francisco, CA, US	permaink
	ACH-SAC	22 - 26 Mar 2000	Lausanne, OI	permalinik

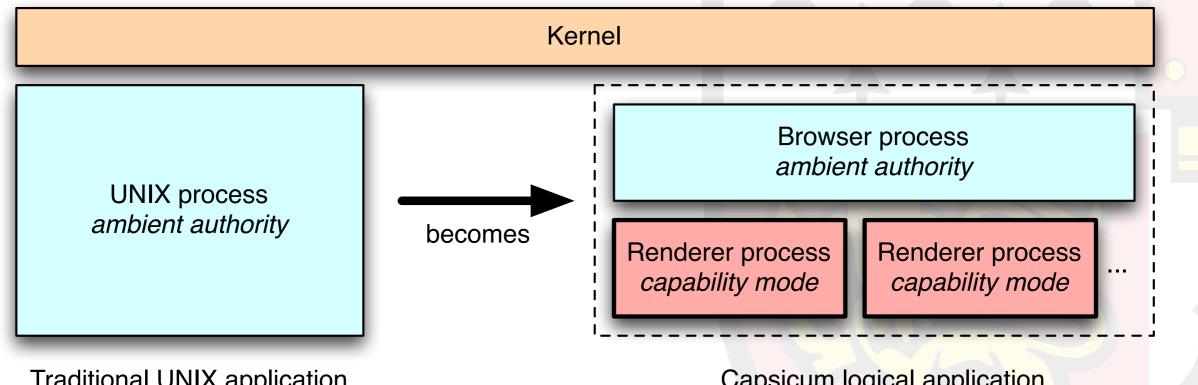
e cinis	https:// mail.google.com /mail/?uhva=L#inbox	8 > 1012
Auf Galendar Documents	finaler lite mon * jonathan.robert.anderson@s	pmail.com Settings Help Sign.o
Google	Search-Mail Search for web Control Mar	
Moore	Contrast.	
Compose Mail		1 - 1 - 1 - 1
Induce	Analyse Report span Doleta More to e Labels e More actions e Refresh	1 - 50 of 2005 Older > Oldest a
But to	Select Al, None, Read, Unread, Stamed, Unstamed	
Starred 🔾	• 0 • • • • • • • • • • • • • • • • • •	10.24
Sert Mail	10 · · · · · · · · · · · · · · · · · · ·	08.43
Drafts (1)	0	08:37
Spam (51)	0	07.46
Bin	10 0	04.32
[imap)/Orafis	0	22 Feb
Junk (GMal)	0	22 Feb
8 moren	0.0	22 Peb
Cortacts	0	22 Feb
Tanks	0.0	22 Feb
0 Owl		22 Feb
Search, add or invite		22 Feb
a burner burner	0	22 Feb
u Jonathan Anderson	1011	22 Feb
	0	22 Feb
You are invisible. Go visible		22 Peb 22 Peb
u Andy Burke		
Christina Anderson		22 Peb 22 Peb
Clark Piller	0	
Denick Lee		21 Feb
 Jonathan Anderson Kayla Johnson 	0	21 Feb
ii Michael Burton	0	21 Feb

CVEs in Jan-Aug 2009				
Firefox	85			
Safari	59			
IE	48			
Chrome	39			
Flash	35			
source; Justin Foster, OWASP				





Logical applications

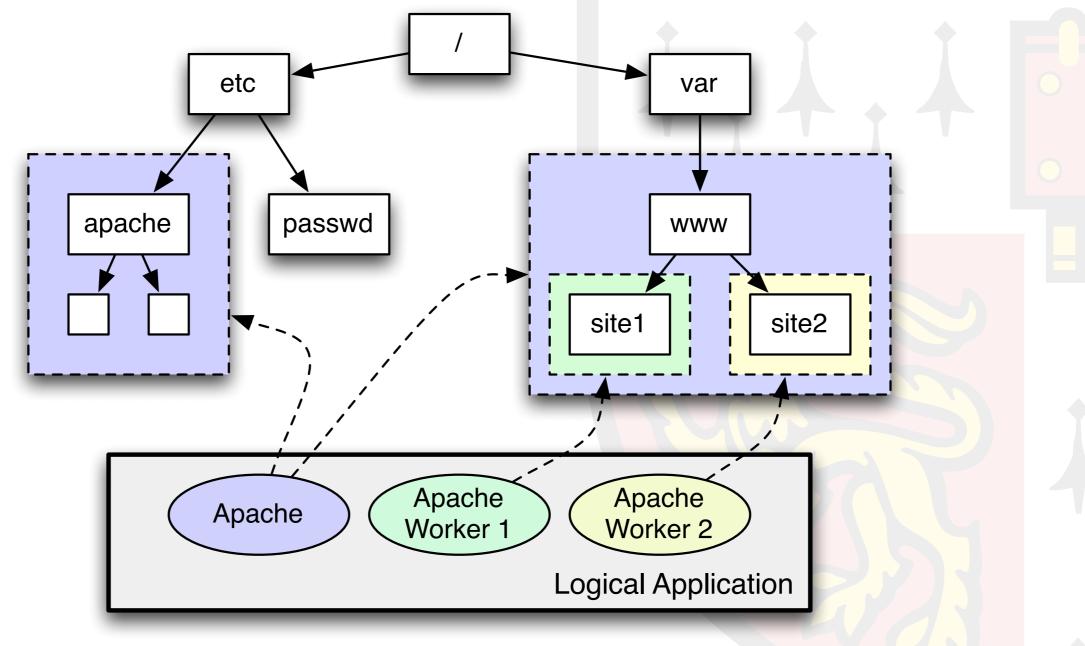


Traditional UNIX application

Capsicum logical application

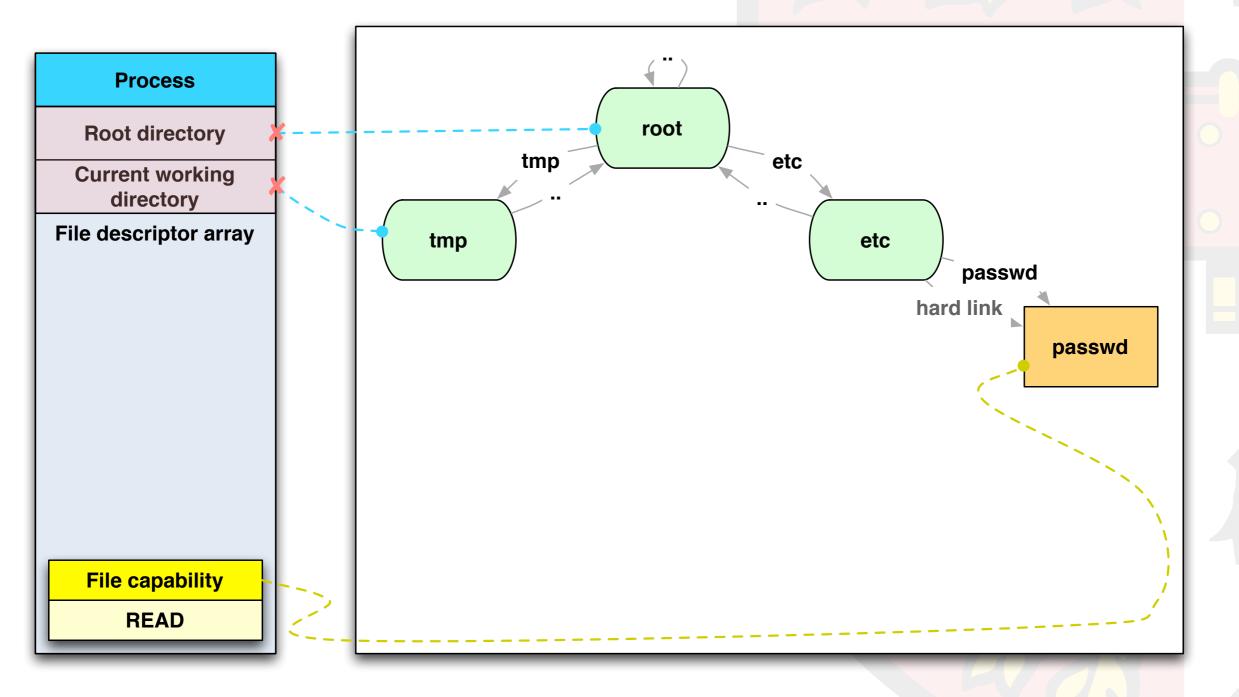


Hierarchical delegation with capabilities





File capabilities





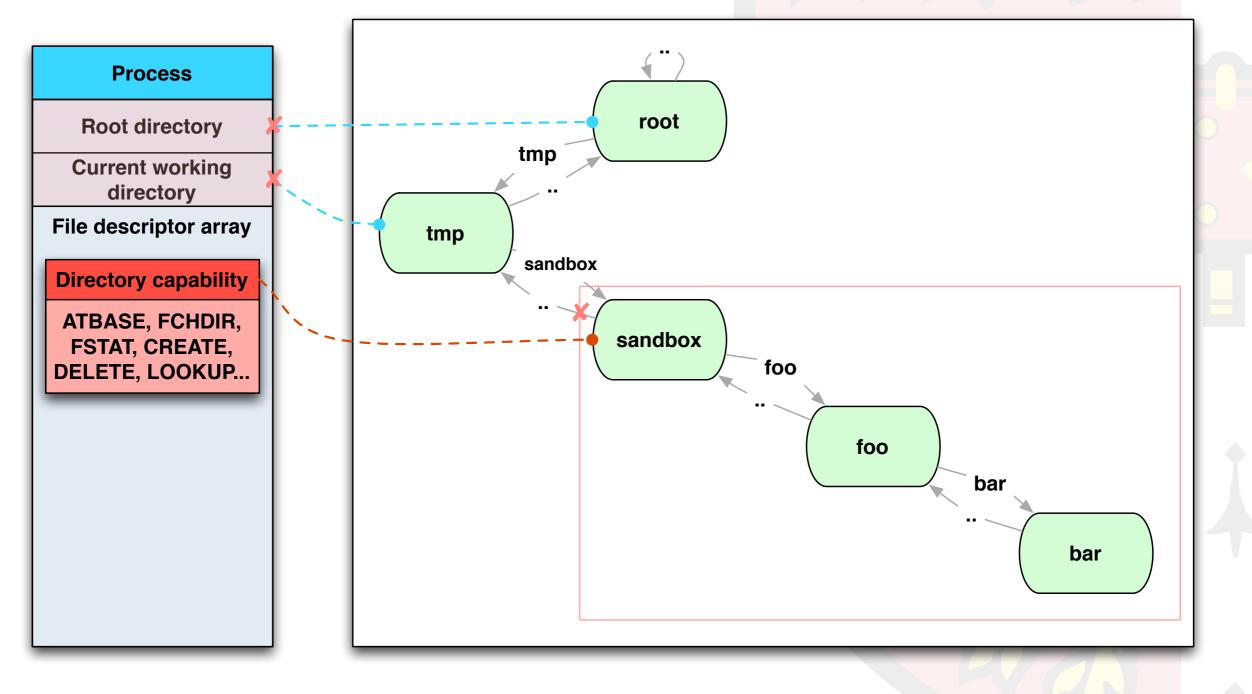
at(2) APIs

• Variations accepting directory descriptors:

- Avoid intermediate lookup state/costs
- Use at(2) calls to delegate directories
- Grant rights to objects **under** capability

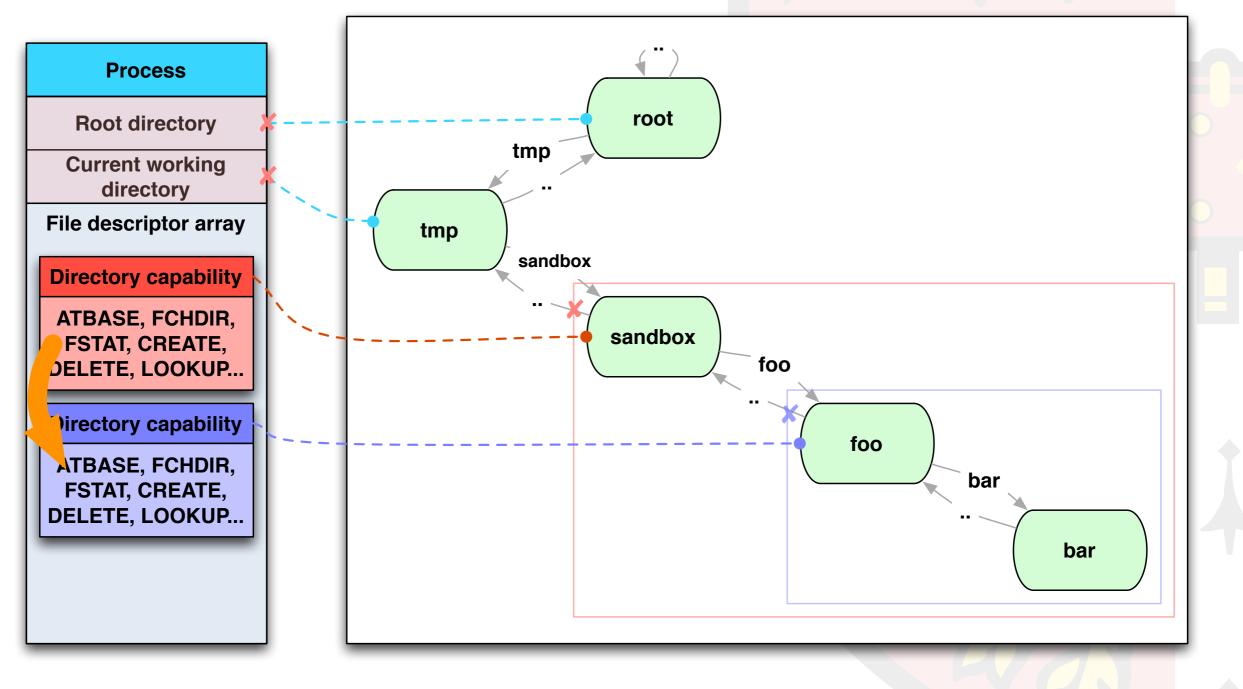


Directory delegation





Derived capabilities



UNIVERSITY OF CAMBRIDGE

Implementing under

- Reject at(2) on absolute paths
- Reuse existing namei lookup code
- Require directory capability argument
- If "..." is looked up relative to starting directory, return ENOTCAPABLE



A concurrency vulnerability



Concurrency

- Multiple computational processes
 execute at the same time and may interact with each other
- Concurrency leads to the appearance of non-determinism



Concurrency vulnerabilities

- When incorrect concurrency management leads to vulnerability
 - Violation of specifications
 - Violation of user expectations
 - **Passive** leak information or privilege
 - Active allow adversary to extract information, gain privilege, deny service...



From concurrency bug to security bug

- Vulnerabilities in security-critical interfaces
 - Races on arguments and interpretation
 - Atomic "check" and "access" not possible
- Data consistency vulnerabilities
 - Stale or inconsistent security metadata
 - Security metadata and data inconsistent

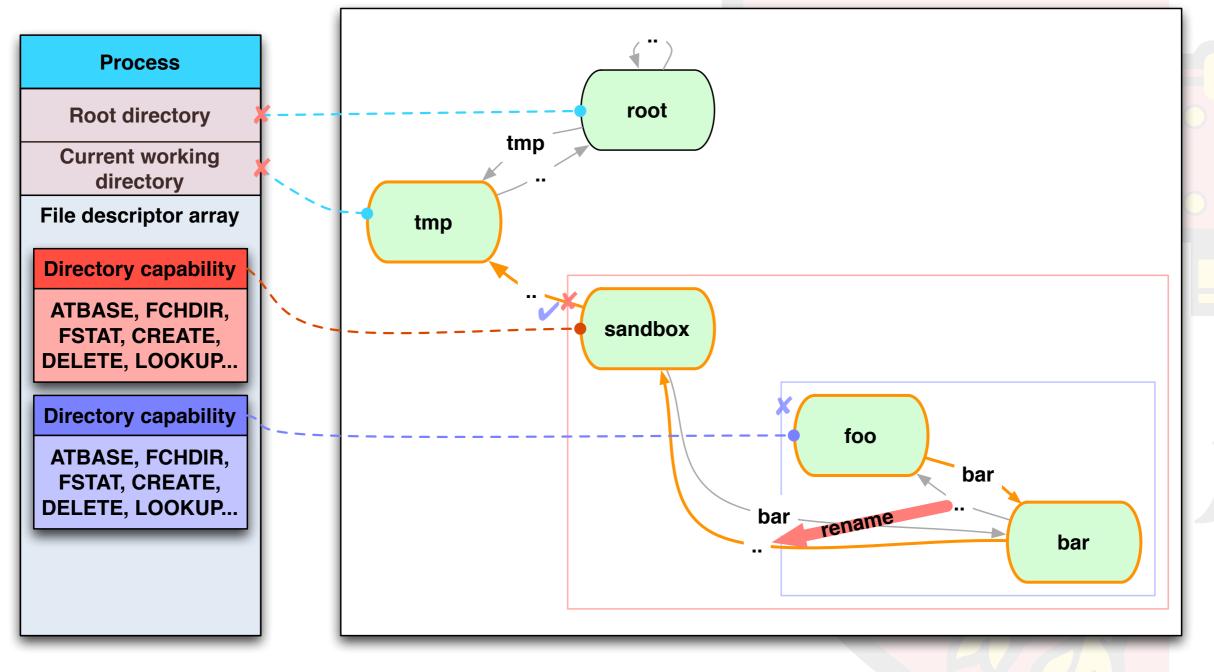


Concurrency attacks on APIs

- System call API bridges "untrusted" userspace and "trusted" kernel
- Attacker's goal to manipulate APIs and trigger security incorrectness in "trusted" implementation
- In software, usually done using multiple client threads/processes and system calls, LPCs, or RPCs to a multithreaded server



openat(foofd, "bar/../.."); renameat(foofd, "bar", sandboxfd, "bar");



UNIVERSITY OF CAMBRIDGE

Concurrency vulnerabilities

- Most race conditions are time-of-check-totime-of-use (TOCTTOU)
- This vulnerability is **not** TOCTTOU
- Bisbey 1978 "unexpected concurrency"
- Security failure due to programmer not understanding concurrency opportunity



The vulnerability

- Bypass containment if any writable directory capabilities passed to sandbox
- Dual-core notebook required ~100,000 loops to exploit
- Exploits non-atomic namespace lookup relative to other operations
 - A performance feature we can't remove!



Why formal methods?

- Serious but subtle concurrency vulnerability with unclear implications
- Namespace containment widely used in UNIX; chroot and beyond
- Want to show that other combinations of name space calls can't trigger similar vulnerabilities



Model checking

- Summary: Clarke, Emerson, Sifakis 2007
 Turing Award lecture; Comm ACM 2009
- Finite state machine represents system under analysis
- Express safety properties in temporal logic
- Exhaustively check model conformance
- Common in protocol, hardware verification



The goal

- Model the relationship between the attacker and the file system implementation
- Want to explore all possible interleavings of events the attacker can trigger
- Validate critical assertions



The model

- Selected SPIN model checker
- 222-line Promela model of system/attacker
- Model a finite set of concurrent processes, each with a limited system call vocabulary
- Finite-sized file system (8 nodes);
 Initial path configuration similar to picture
- Assertion: multi-".." lookups fail



Solution space

Approach	Performance	Functionality	Security
Remove subtree delegation	~	×	~
Namespace walk	×	~	×
Limit namespace concurrency	×	~	✔ (NFS: ¥)
Limit""		×	~

Limitations

- Hand-crafted Promela mode significantly different semantics and implementation from kernel code
- Finite process count
- Limited system call vocabulary
- Limited file system size
- Want stronger "can't name root" assertion



Conclusion

- Capsicum: practical capabilities for UNIX
- Concurrency vulnerability with serious real-world implications for Capsicum
- Applied model checking
- Improved our confidence in security / performance / functionality trade-off





