A Practical Dynamic Buffer Overflow Detector

Olatunji Ruwase , Monica S. Lam

Presented by:Erinc Arikan

What's This Paper About?

- Dynamic Buffer Overflow Detectors
- CRED(C Range Error Detector)
- Results of the tests that are done by CRED.
- sOME

Buffer Overflows

- Still being discovered in programs in use.
- In year 2002 57% of CERT advisories were related to it.
- In August 2003 50% of security advisories fell under it.

Buffer Overflows

- Can cause billions of dollars worth damage in the computing community.
- Recent worms used this vulnerability.
- Some Examples: Slammer , CodeRed , Blaster , Welchia

Buffer Overflow Attacks

- Attacker tries to modify the memory state to control the program in the priviliged mode.
- Example:Smashing the stack.

Smashing the Stack

- Attacker may overwrite the return address of the function on the stack.
- Attackers may modify the locations referenced by function pointer.
- Attackers may modify the global offset table

Detecting Buffer Overflows

- A practical solution could not be found for years.
- Static Detectors vs Dynamic Detectors

Static Buffer Overflow Detectors

- Try to verify that all memory accesses are checked for overruns.
- Disadvantages:
  - The problem is undecidable .
  - Sound tools may genarate false warnings .
  - Unsound tools may miss vulnerabilities .
  - Requires manual inspection and fix of the code .

Dynamic Buffer Overflow Detectors

- They automatically insert the necesaary guards.
- It must also:
  - Protect against all buffer overflow attacks .
  - Not break working code .
  - Be efficient .

Dynamic Buffer Overflow Detectors

- Some of them only offers protection against smash stacking.
  - Examples:StackGuard , StackShield, Propolice
- Some replaces the pointers with a structure that will be used for bounds checking.

Dynamic Buffer Overflow Detectors(Continued)

- Jones and Kelly proposes another model by using referent objects.
- It has some weaknesses.
  - False warnings
  - Computational Overhead

Bounds Checking Using Referent Objects

- Proposed by Jones and Kelly.
- It states that address computed from an in bound pointer must share the same referent object as the original pointer.
- It uses an object table which is stored as splay tree.

Handling Out of Bounds Pointers

- Jones and Kelly approach pads each object by 1 byte.
- All out of bounds addresses are replaced by ILLEGAL value.
- What if we store an out of bound address,and then use it again to compute an in bound address.

Proposed Solution for Out of Bounds Pointers(CRED)

- Create a OOB (Out of Bounds) object for every out of bounds address value.
- OOB Object includes:
  - Out of Bounds address value.
  - Referent Object that value refers to
- OOBs are stored in OOB Hash Table.

Proposed Solution for Out of Bounds Pointers(CRED)

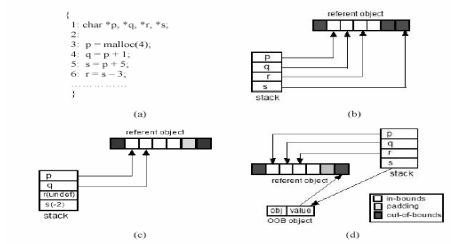
- Procedure
  1. If the address computed is Out of Bounds,it is replaced by OOB object and hash table is updated.
  2. When a pointer is derefenced,if it is not pointing to an unchecked object or to an object in object table,abort and print error message.
  3. If a pointer is used in arithmetic/comparison operation,and if it is not pointing to unchecked object or to an object in the object table,value is retrieved by looking OOB hash table.

Proposed Solution for Out of Bounds Pointers(Continued)

- Procedure:

  4. When an object is deleted,delete all the OOBs referring to it,update the OOB hash table.

Example



Run Time Overhead

- Performance of Jones and Kelly's technique was another disadvantage.
- Generally buffer overflow attacks uses string data.
- Also when data is copied between memory locations it's in the form of char *
- So bounds checking can be restricted to strings.

Experiments with CRED

- CRED is merged with the Jones and Kelly checker for gcc 3.3.1
- It's used to test the 20 common open source programs.
- It's run by checking all buffers,not only strings

Results of Experiment

Program	Type	# Lines	Vuln.	Tests	JK	CRED
Apache-1.3.24	web server	73.6K	no	yes	fail	pass
binutils-2.13.2.1	binary tools	596.5K	no	yes	fail	pass
bison-1.875	parser generator	25.1K	no	yes	fail	pass
ccrypt-1.4	encryption utility	4.4K	no	yes	pass	pass
coreutils-5.0	file, shell, & text utilities	69.5K	no	yes	fail	pass
enscript-1.6.1	ascii to postscript converter	22.1K	no	yes	fail	pass
gawk-3.1.2	string manipulation tool	36.4K	yes	yes	fail	pass
gnupg-1.2.2	OpenPGP implementation	71.2K	no	yes	fail	pass
grep-2.5.1	pattern matching utility	20.8K	no	yes	fail	pass
gzip-1.2.4	compression utility	5.8K	yes	yes	pass	pass
hypermail-2.1.5	mail to HTML converter	27.6K	yes	yes	fail	pass
monkey-0.7.1	web server	2.5K	yes	no	pass	pass
OpenSSH-3.2.2p1	SSH1 protocol implementation	43.4K	no	no	fail	pass
OpenSSL-0.9.7b	SSL & TLS toolkit	162.7K	no	yes	fail	pass
pgp4pine-1.76	mail encryption tool	3.3K	yes	no	fail	pass
polymorph-0.40	filesystem unixier	0.4K	yes	no	pass	pass
tar-1.13	archiving utility	18.2K	no	yes	pass	pass
WsMp3-0.0.10	web server	3.4K	yes	no	pass	pass
wu-ftpd-2.6.1	FTP server	18.3K	no	no	pass	pass
zlib-1.13	data compression library	8.3K	no	yes	pass	pass

Some Comments on Results

- CRED passed all the tests.
- Jones and Kelly failed %60 of the tests.
- CRED also found some previously unknown bounds errors.

Protection Experiments

1. 7 Known vulnerable programs are instrumented with CRED.An attacks are applied to them.
2. 20 different buffer overflows attacks are used to test the CRED.

Results of the Performance Tests

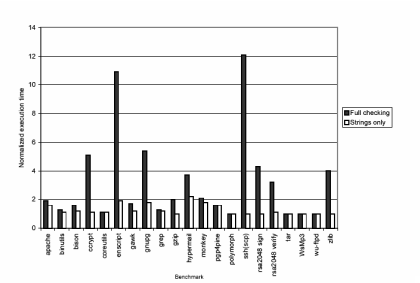
- To determine the overhead of CRED because out of bounds adress tracking, we compare it with the Jones and Kelly technique

	JK(s)	CRED (s)
ccrypt	26.93	23.00
gzip-1.2.4	0.19	0.18
monkey-0.7.1	5.60	6.00
polymorh-0.4.0	0.39	0.39
tar-1.13	0.66	0.76
WsMp3-0.0.10	1.48	1.48
wu-ftpd-2.6.1	33.40	33.40
zlib-1.13	0.11	0.11

Evaluation Criteria for 8 Programs

Program	What was evaluated
Apache-1.3.24	Response time to 15K tcp connections at the rate of 90 per second.
monkey-0.7.1	Response time to 3K tcp connections at the rate of 50 per second.
openssh-3.2.2p1	Latency of 126MB file transfer using scp via the network loop back interface.
OpenSSL-0.9.7b	Time to sign and verify 2048 bit keys using rsa.
pgp4pine-1.76	Time to decrypt 1MB file
polymorph-0.40	Time to convert names of 100 files to unix style (lower case) names.
WsMp3-0.0.10	Latency of downloading a 1.5MB file.
wu-ftpd-2.6.1	Latency of 126MB file transfer via the network loop back interface

Performance Experiments



Related Work

- Static Analysis Approaches
  - Wagner proposed a system where overflow detection system proposed as an integer constraint problem.
  - Larochelle and Evans presented Annotation asisted system.
  - CSSV(C String Static Verifier)detects string manipulation errors by the help of procedure summaries.

Related Work

- Dynamic Analysis Approaches
  - StackGuard,Propolice,StackShield,Baratloo are programs that prevents stack smashing attacks.
  - Lhee and Chapin presented a buffer overflow technique using array bounds checking.
  - STOBO is a tool that can find vulnerabilities due to use of library functions.

Related Work

- Mixed Approach
  - Ccured,Cyclone first statically detects the buffer overflows and then runs runtime tests.
  - DynamoRIO uses program shepherding to detect buffer overflows/

Conclusions

- CRED is a practical dynamic overflow detector for C programs.
- It is built on Jones Kelly technique.
- It is more efficient and succesful with respect to other methods.
- Further improvement is possible with bounds checking and using static checking beforehand.

Questions?