Address Space Layout Randomization (ASLR) Effective or Not?

Radu-Constantin Onuțu

University of Bucharest

Introduction

- **Reference**: Hovav Shacham, Matthew Page, Ben Pfaff, Eu-Jin Goh, Nagendra Modadugu, and Dan Boneh. On the effectiveness of address-space randomization. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 298–307, 2004.
- What is ASLR and why is it important?
- Is ASLR an effective security measure?

Introduction

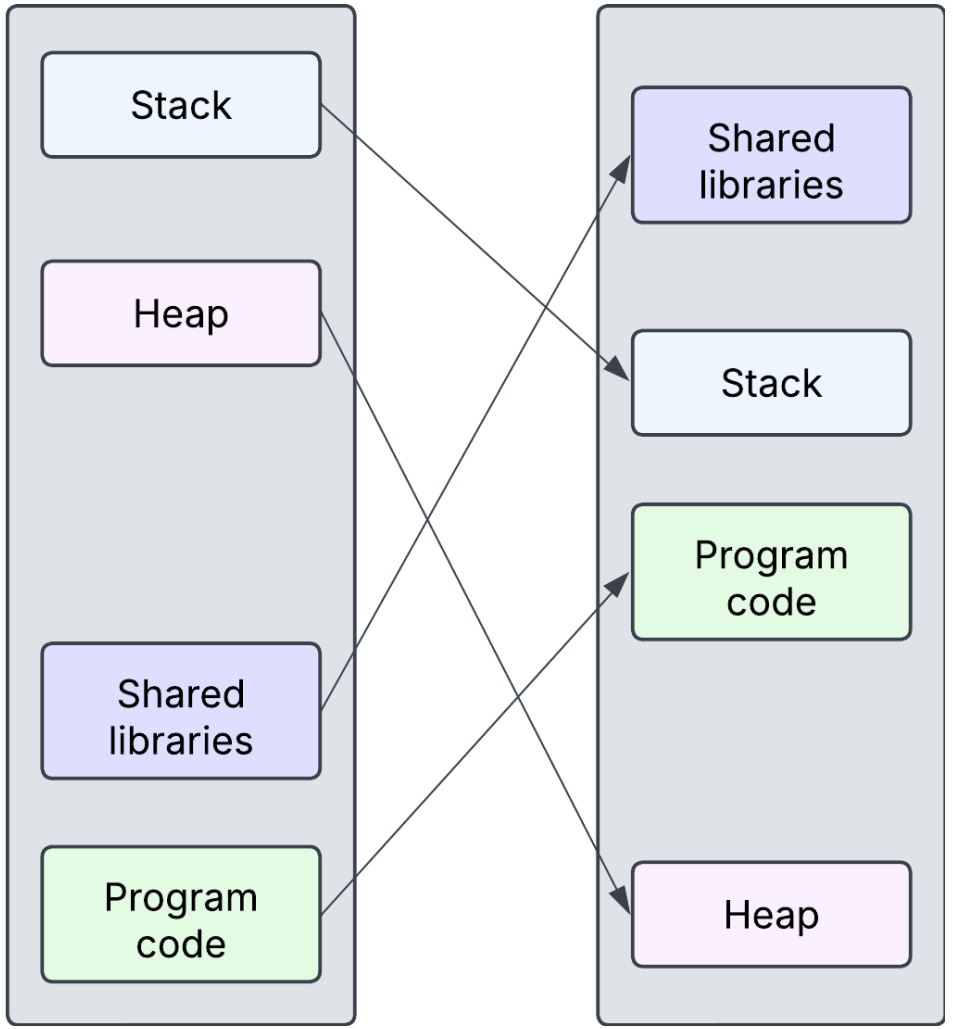
- **Reference**: Hovav Shacham, Matthew Page, Ben Pfaff, Eu-Jin Goh, Nagendra Modadugu, and Dan Boneh. On the effectiveness of address-space randomization. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 298–307, 2004.
- What is ASLR and why is it important?
- Is ASLR an effective security measure?

Introduction

- **Reference**: Hovav Shacham, Matthew Page, Ben Pfaff, Eu-Jin Goh, Nagendra Modadugu, and Dan Boneh. On the effectiveness of address-space randomization. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 298–307, 2004.
- What is ASLR and why is it important?
- Is ASLR an effective security measure?

What is ASLR?

- ASLR is a security technique that aims to prevent memory corruption attacks by randomizing the locations of key memory regions.



Why is ASLR important?

- ASLR ensures that the likelihood of attack through buffer overflows and finding the space address of applications become more difficult.

```
radu@DESKTOP-0SLLE3M:~$ cat main.c
#include <stdio.h>

int main() {
        int var = 0;
        printf("Address of var: %p\n", &var);
        return 0;
}
radu@DESKTOP-0SLLE3M:~$ ./main
Address of var: 0x7ffce238b7f4
radu@DESKTOP-0SLLE3M:~$ ./main
Address of var: 0x7ffd0f641ed4
```

Shacham *et al.'s* Study

- Studied ASLR on PaX ASLR Linux.
- Targeted Apache web server using a buffer overflow

Results:

- ASLR is weak on 32-bit systems due to low entropy (only 16 bits for randomization i.e.: $2^{16} = 65.536$ probes).
- Brute-force attack bypassed ASLR in 216 seconds on average.

Shacham *et al.'s* Study

- Studied ASLR on PaX ASLR Linux.
- Targeted Apache web server using a buffer overflow

Results:

- ASLR is weak on 32-bit systems due to low entropy (only 16 bits for randomization i.e.: $2^{16} = 65.536$ probes).
- Brute-force attack bypassed ASLR in 216 seconds on average.

Results of the study

Time to bypass ASLR:

Average	**Min**	**Max**
216s	29s	810s

Traffic generated:

Average	**Max**
6.4 MB	12.8 MB

Limitations of "Watchers"

In the paper, the authors talk about a crash detection system called a "watcher".

- If the *watcher* alerts an administrator then the administrator cannot react in time (diagnose the network traffic, asses the severity and take corrective measures in only 216 seconds).
- If the *watcher* shuts down the daemon while waiting for intervention this could introduce a risk for denial-of-service attacks.

Total cost of one hour of downtime

Patterson, David A. "A Simple Way to Estimate the Cost of Downtime." LISA. Vol. 2. 2002.

Method	Dictionary
Brokerage operations	$6,450,000
Credit card authorization	$2,600,000
Ebay	$225,000
Amazon.com	$180,000
Package shipping services	$150,000
Home shopping channel	$113,000
Catalog sales center	$90,000
Airline reservation center	$89,000
Cellular service activation	$41,000
On-line network fees	$25,000
ATM service fees	$14,000

Conclusion

Is ASLR an effective security measure?

- ASLR alone is **not** good enough since on its own. It fails against brute-force attacks (on 32-bit architectures).
- Its effectiveness improves on 64-bit architectures and when combined with other defense measures (stack canaries, W$\oplus$X, etc.).

Thank You!