Buffer Overflow Exploiting Basics

3. Kassel Code Meetup

Sebastian Brabetz – 10.09.2014

Overview

1. Introduction
2. Buffer Overflow Theory
3. Live Demo
4. More Information
5. Questions

whoami

- Worked 4 years as Security- / Firewall-Administrator
- 1 year as IT Security Specialist
- Currently IT Security Engineer
- OSCP since 06/2014

- Blog: http://itunsecurity.wordpress.com
- Email: sebastian.brabetz@web.de
- IRC: irc.hackint.org / #ccc-ks
- Twitter: @teh_warriar
- Feel free to contact me with questions!

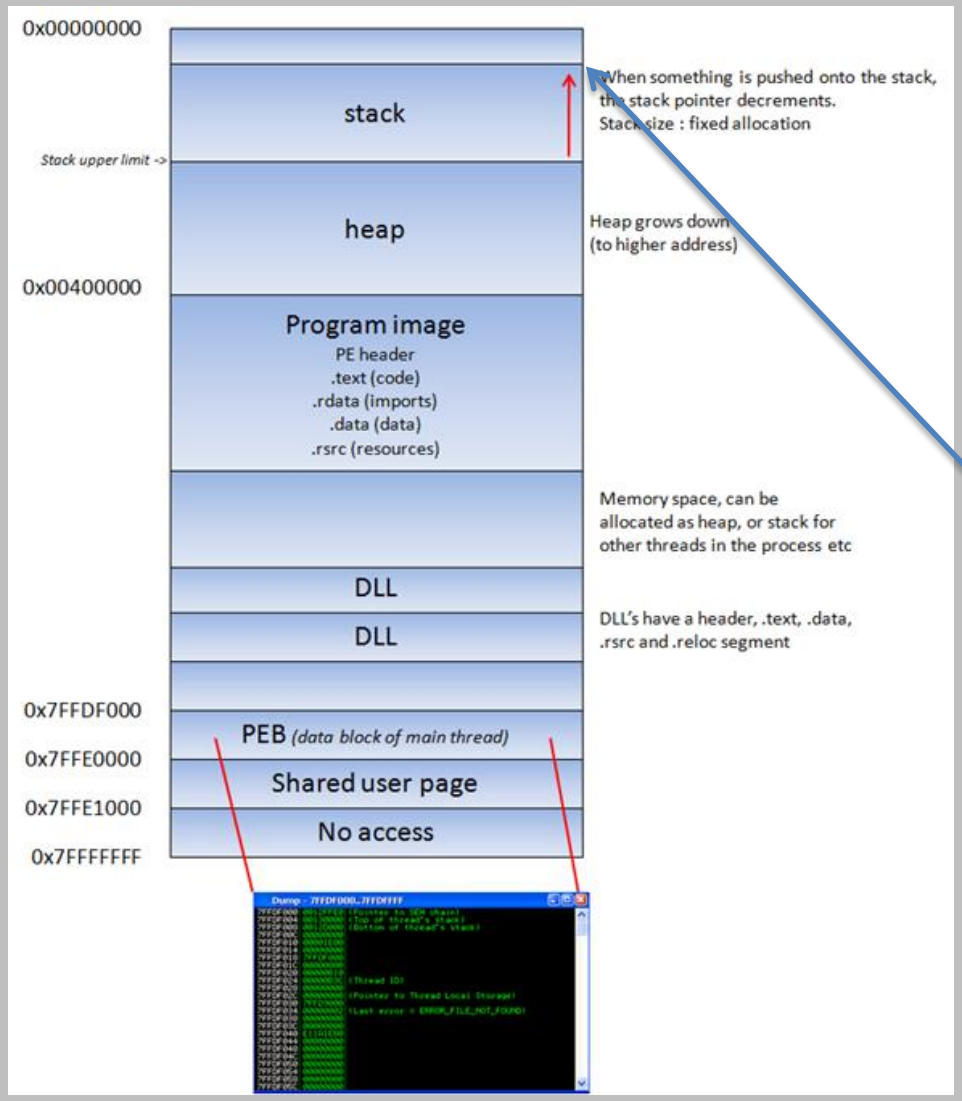
What is a (stack) Buffer Overflow?

Wikipedia:

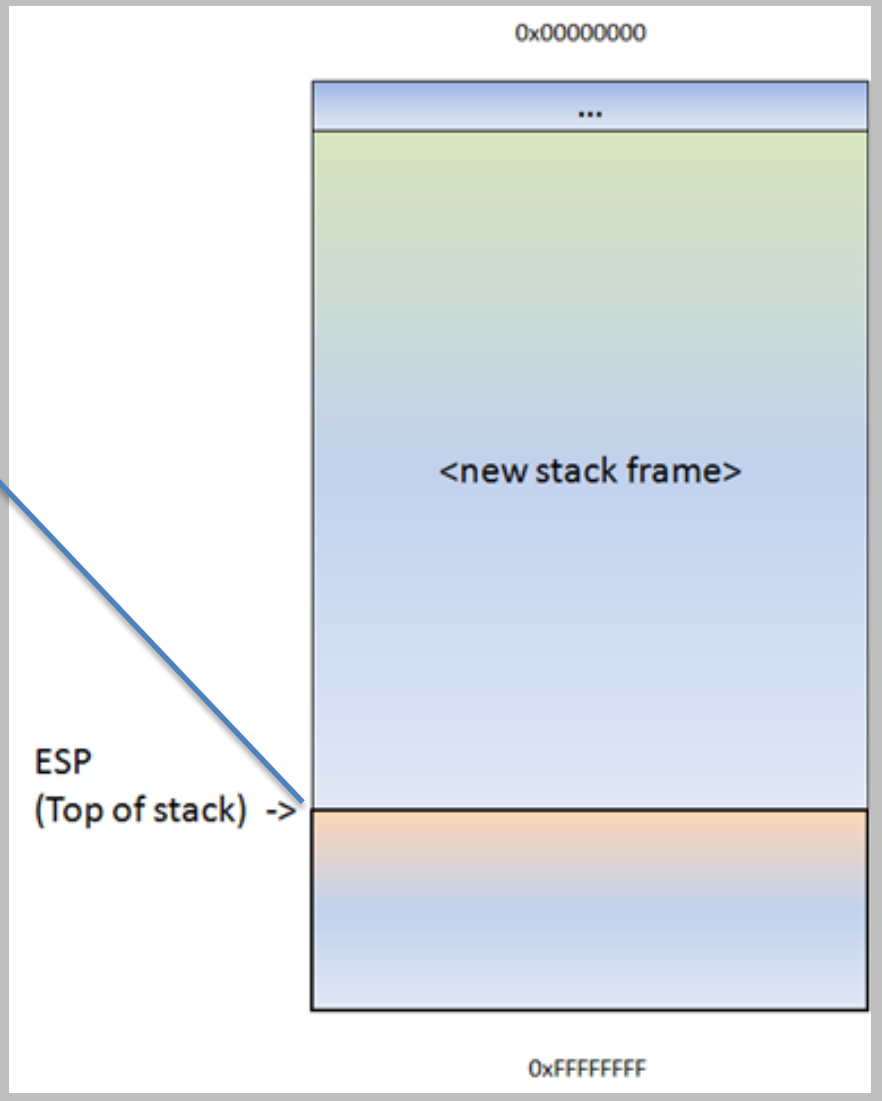
In software, a stack buffer overflow or stack buffer overrun occurs when a program writes to a memory address on the program's call stack outside of the intended data structure; usually a fixed length buffer.

What is a (stack) Buffer Overflow?

win32 process memory map



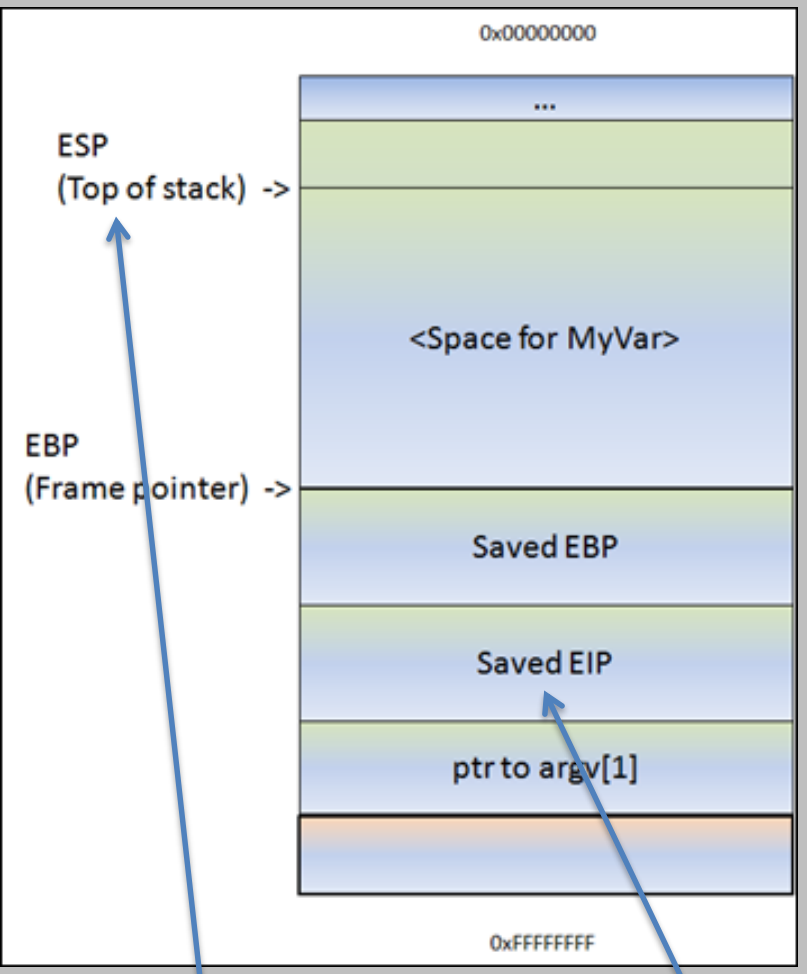
new stack frame will be placed on top of the stack



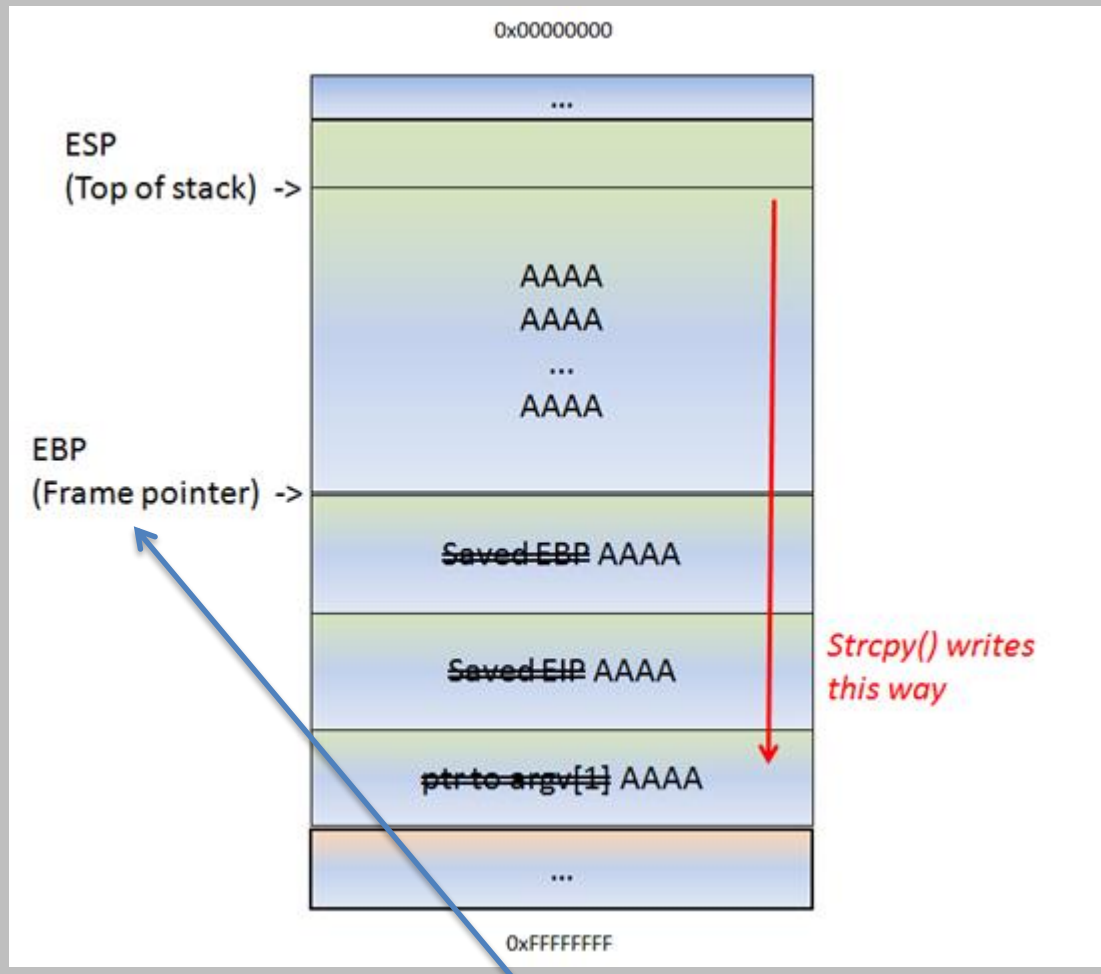
Illustrations from http://www.corelan.be

What is a (stack) Buffer Overflow?

stack after function prologue



buffer overflows into EIP



ESP always points to the top of the current stack context

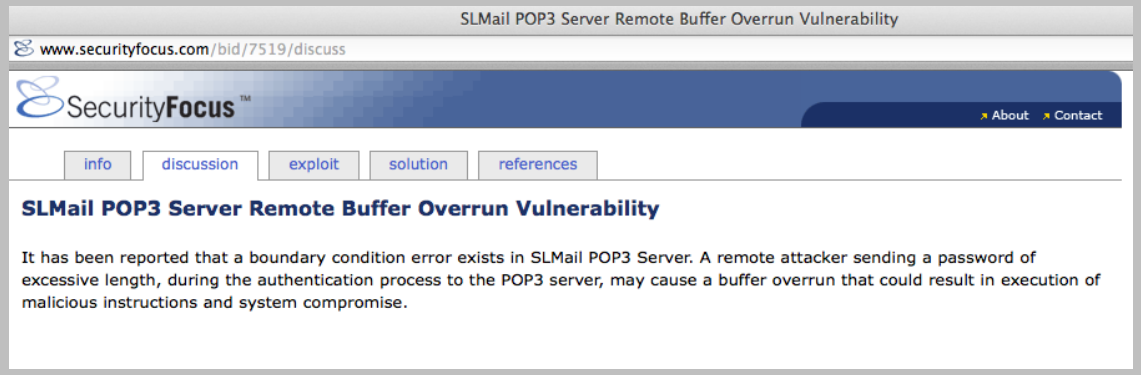
The EIP (saved during the function prologue) tells the CPU where to return after the function (gets loaded into the CPU during the function epilogue)

EBP always points to the base of the current stack context

Illustrations from http://www.corelan.be

Live Demo: The Vulnerable Program

SeattleLab Mailserver: SLMail Version 5.5.0.4433



Great for demonstrations because:

- POP3 is an easy Protocoll
- Cleartext Protocoll
- Simple Stack Overflow vulnerability

„A remote attacker sending a password of exessive length, during the authentication process to the POP3 server, may cause a buffer overrun that could result in execution of malicious instructions and system compromise.“

Live Demo

What could possibly go wrong?!

More information / Links

What you need to reproduce this:

- Windows XP SP2 German
  (other versions might have different jmp esp adresses!)
- Immunity Debugger (see next slide)
- Mona.py (see next slide)
- SLMail 5.5.0.04433
  http://slmail.software.informer.com/5.5/
- Kali Linux http://www.kali.org
- **This slides and all scripts are available for download on the codemeetup website right now!**

More information / Links

Python Socket Programming:

- http://openbook.galileocomputing.de/python/python_kapitel_20_001.htm

Learning Program Vulnserver:

- http://www.thegreycorner.com/2010/12/introducing-vulnserver.html
- http://rockfishsec.blogspot.de/2014/01/fuzzing-vulnserver-with-peach-3.html
- http://resources.infosecinstitute.com/fuzzing-vulnserver-discovering-vulnerable-commands-part-1/

Free Exploit Writing Tutorials from Corelan:

- https://www.corelan.be/index.php/articles/
- https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

11 Part Tutorial including SHE, DEP and ASLR Bypassing, Egg Hunting, ROP and more!

Immunity Debugger and monay.py Plugin from Corelan:

- http://debugger.immunityinc.com
- https://www.corelan.be/index.php/2011/07/14/mona-py-the-manual/

German Book about Buffer Overflows on Amazon:

- http://www.amazon.de/Buffer-Overflows-Format-String-Schwachstellen-Funktionsweisen-Gegenmaßnahmen/dp/3898641929/ref=sr_1_1?ie=UTF8&qid=1410008572&sr=8-1&keywords=buffer+overflow+und+format+string+schwachstellen
- Free Look inside 1st Chapter (explains Stack Memory Management!): https://www.dpunkt.de/leseproben/2003/Auszug_aus_Kapitel_1.pdf

Questions?

Backup

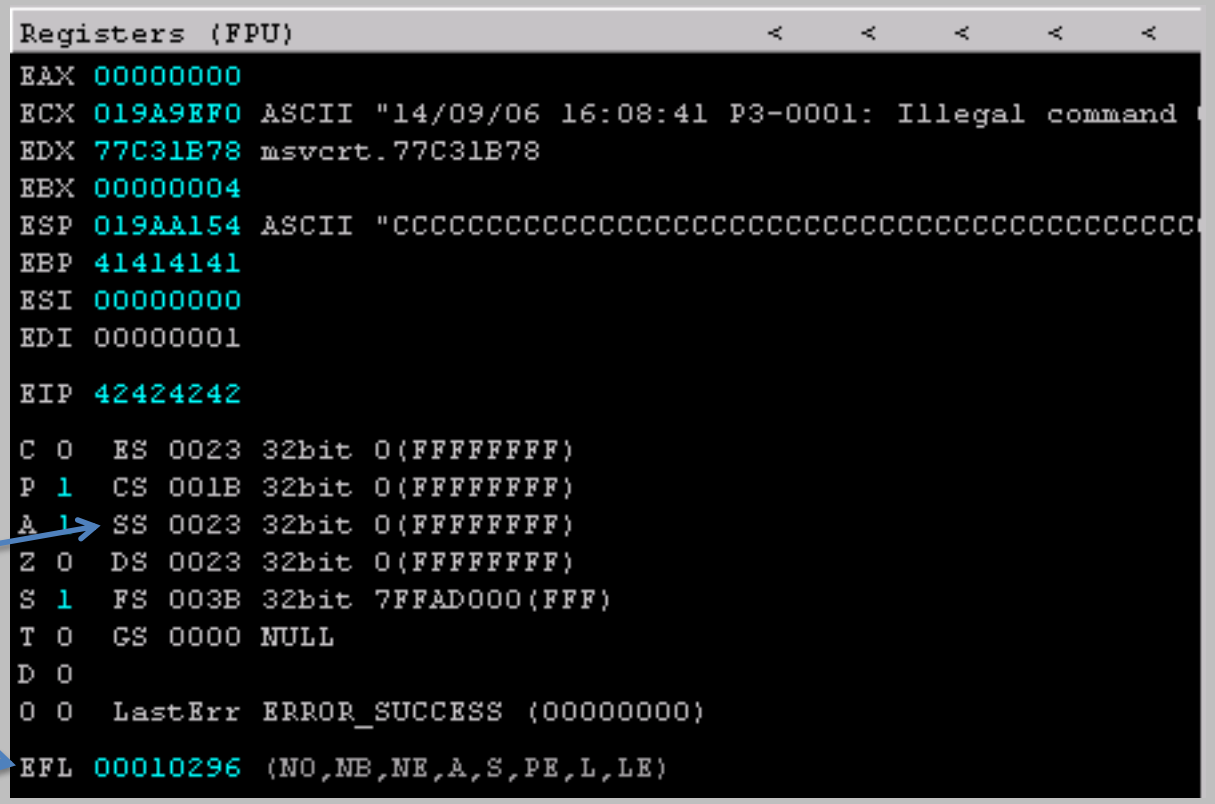
CPU Register

8 General purpose registers for arythmetic operations and string operations as well as stack and base pointer to mark the stack context

Extended Instruction Pointer points to the next instruction the CPU will exectue
Not writeable from the program!

6 Segment registers to identify segments in the memory

EFLAGS Register can store and provide status informations about the running program



Stack

Abb. 1–1
Typisches Prozess-speicher-Layout eines C-Programms