Binary Exploitation

Talking with binaries — 0x1

Roadmap

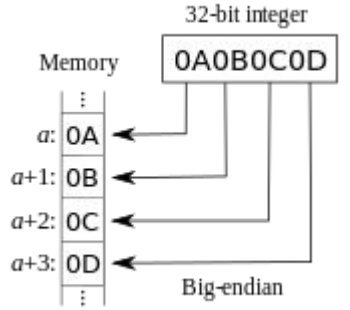
- Motivation
- Endianness
- Interaction
  - Command Line 101
  - Using Python
  - Examples

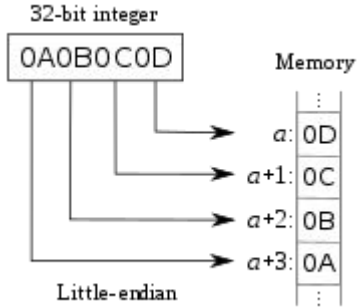
Motivation

- Large inputs
- Non ASCII characters
- Limited interaction time
- Encoding/Decoding

Endianness

- How bytes are read/written in memory
  - Big-endian
    - Most significant byte first (big end)
  - Little-endian
    - Least significant byte first (little end)
- Little-endian most common





Endianness

For any data type larger than 1 byte (smallest addressable unit of memory) endianness must be taken into account, e.g.:

- Program reads string and address is to be written
- Program reads integer and string is to be written

Interaction - Command Line 101

- Interaction based on stdin(0), stdout(1), stderr(2)
  - Redirect stdin

```
$./exe <input OR 0<input
```

  - Redirect stdout

```
$./exe >output OR 1>output
```

  - Redirect stderr

```
$./exe 2>outputerr
```

Interaction - Command Line 101

- Redirect stdout and stderr

```
$./exe 2>&1 > output
```

- Arguments

```
$./exe arg1 arg2 … argn
```

- Command substitution (execute and use its output)

```
$echo “Today is $(date).” OR “Today is `date`.”
```

Interaction - Command Line 101

- Pipes

```
$echo “Hello” | ./exe OR ./exe < <(echo “Hello”)
```

Interaction - Using Python

- Simple programming language
- Plenty of libraries (pwntools)
- Widely used

Interaction - Examples

- Use output of python program as input

```
$python program.py | ./exe
```

- Use output of python program as argument

```
$./exe “$(python program.py)”
```

- Use inline python

```
$python -c “import sys; sys.stdout.write(‘A’ * 15)” | ./exe
```

References

http://www.tldp.org/LDP/abs/html/index.html

https://docs.python.org/3/

https://docs.pwntools.com/en/stable/

Next week...

- Buffer Overflows
  - Protostar exercises from exploit-exercises.com