University of BRISTOL

Computer System B -Security

Introduction to Software Vulnerabilities Part 4 Format String

Sanjay Rawat

Format String Errors

- Formatted output functions consist of a format string and a variable number of arguments.
- The format string provides a set of instructions that are interpreted by the formatted output function.
- By controlling the content of the format string a user can control execution of the formatted output function.
- Caller cleans the stack (variadic function).

Example functions

- `vfprintf()`
  - `vprintf()`
  - `vsprintf()`
    - `vsnprintf()`

- `fprintf()`
- `printf()`
- `sprintf()`
- `snprintf()`

Format strings

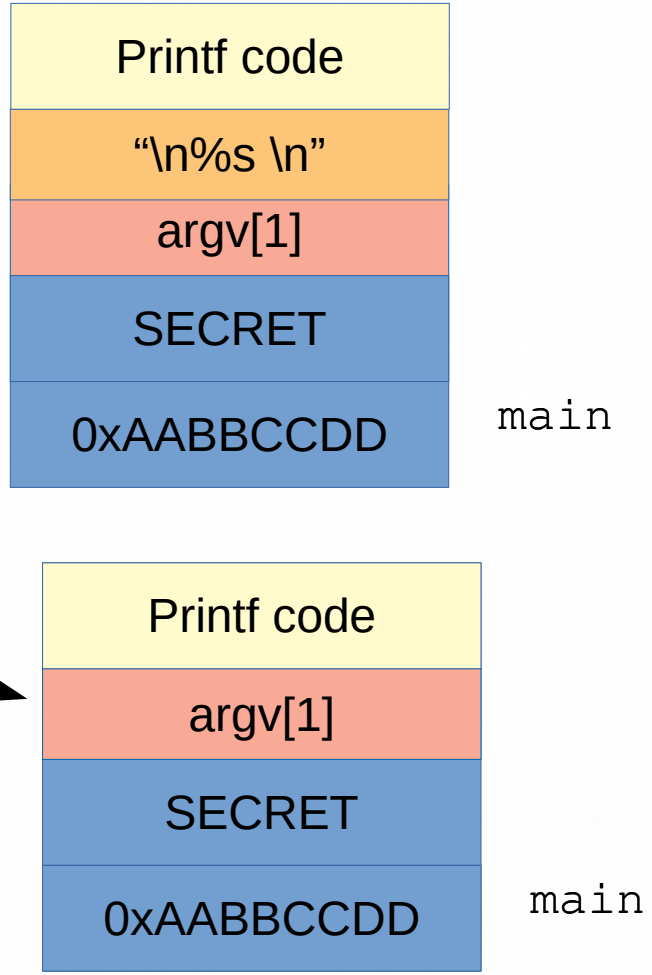
- Format strings are character sequences consisting of *ordinary characters* (excluding %) and *conversion specifications*.
- Conversion specifications convert arguments according to a corresponding conversion specifier, and write the results to the output stream.
- Conversion specifications begin with a percent sign (%) and are interpreted from left to right.
- If there are more arguments than conversion specifications, the extra arguments are ignored.
- If there are not enough arguments for all the conversion specifications, the results are undefined.

Example code

```
#include <stdio.h>
int main(int argc, char **argv)
{
    int i=0xAABBCCDD;
    char secret[]="SECRET";
    if(argc !=2)
    {
        printf("Run with an argument..\
n");
        return -1;
    }

    printf(argv[1]);//insecure call :(
    printf("\n%s\n", argv[1]);// good !!
:D

    return 0;
}
```

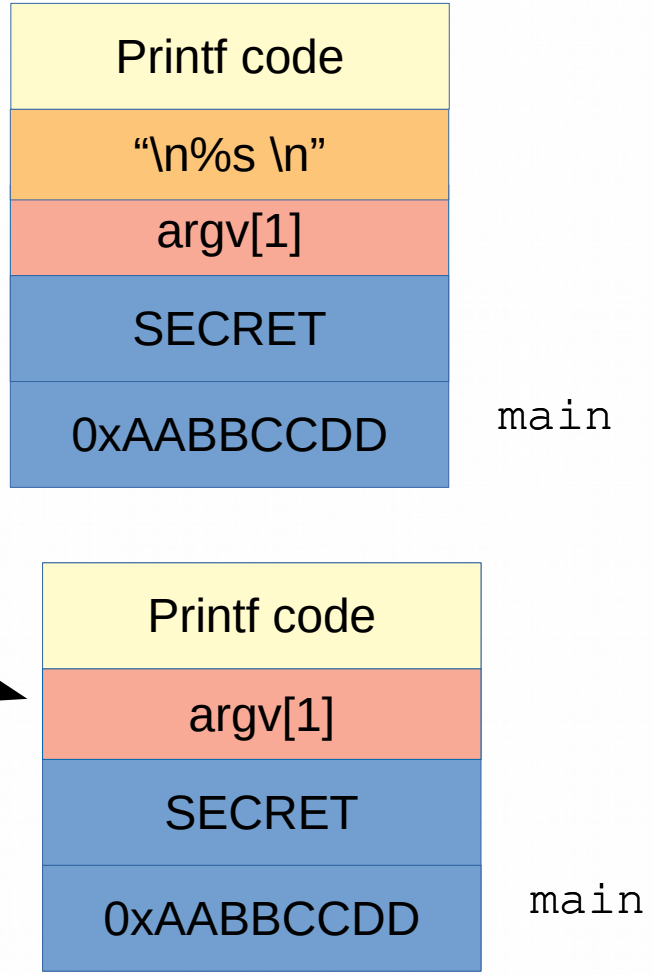


Example code

```
#include <stdio.h>
int main(int argc, char **argv)
{
    int i=0xAABBCCDD;
    char secret[]="SECRET";
    if(argc !=2)
    {
        printf("Run with an argument..\
n");
        return -1;
    }

    printf(argv[1]);//insecure call :(
    printf("\n%s\n", argv[1]);// good !!
:D

    return 0;
}
```



Lets check its assembly too!