Format String Bug

Format String

- printf(“%x”,a);
- printf(“%d %d”,a,b);
- printf(“%c”,&a);
- printf(“%s”,a);

- %x, %d, %c, %s <= Format String

Format String Attack

- Format String에서 일어나는 버그를 이용하여, 원하는 위치에 데이터를 read/ write할 수 있는 공격기법

- 프로그래머들의 실수로 인해 생기는 취약점

Format String Attack

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's 2021");
}
```

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's %d",a);
}
```

Format String Attack

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's 2021");
}
```

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's %d",a);
}
```

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's %d");
}
```

Format String Attack

```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's %d",a);
}
```

printf 함수의 매개변수

rdi : “It’s %d”
rsi : a (2021)

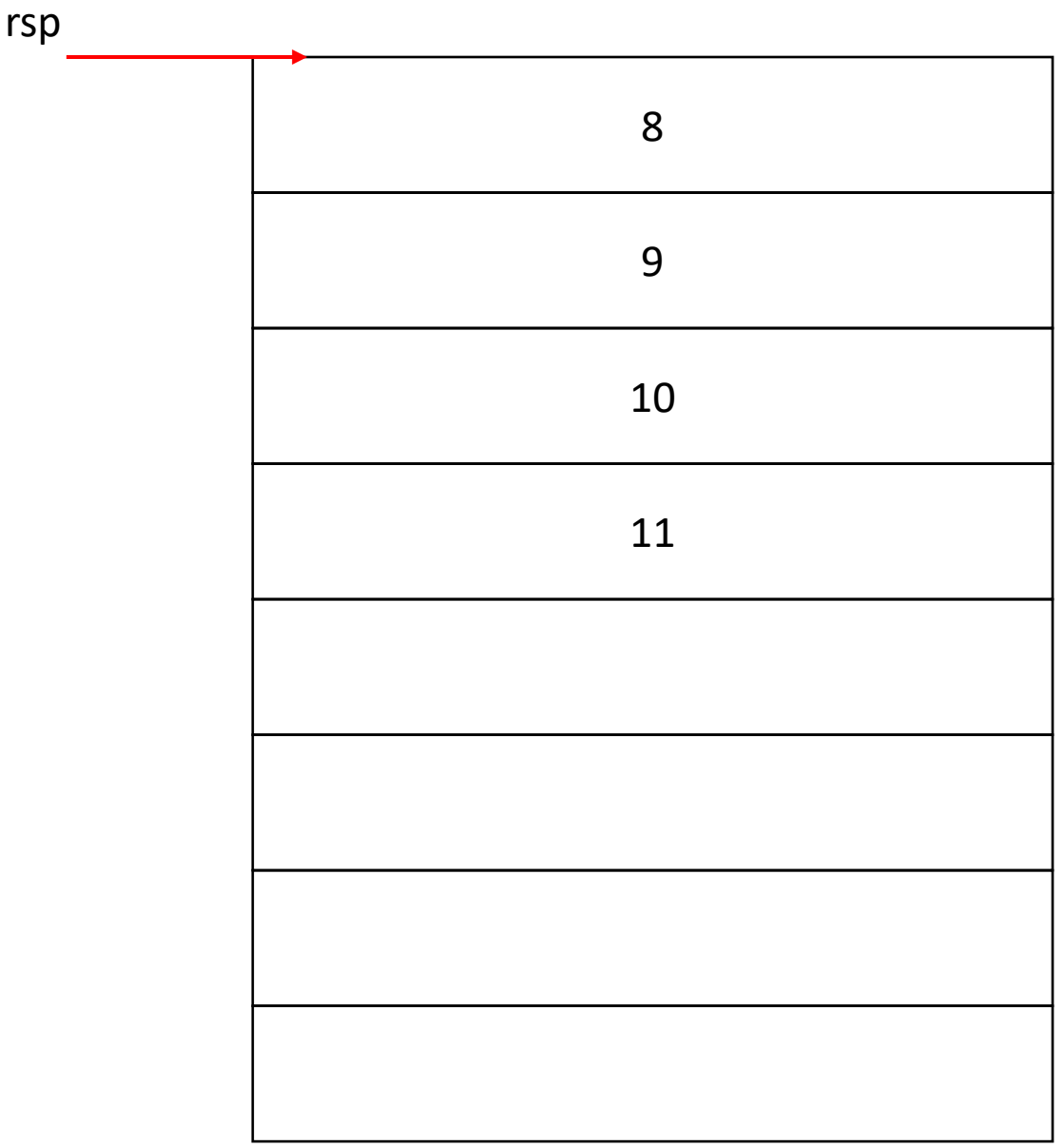
```c
#include <stdio.h>

int main(void)
{
    a = 2021;
    printf("It's %d");
}
```

printf 함수의 매개변수

rdi : “It’s %d”
rsi : ??

현재 레지스터 상태

rax : 1
rbx : 2
rcx : 3
rdx : 4
rsi : 5
rdi : “%x%x%x%x%x%x%x” (7개)
r8 : 6
r9 : 7



Format String Attack

```c
#include <stdio.h>

int main(void)
{
    printf("Hi my name is 1nteger_c");

}
```

```c
#include <stdio.h>

int main(void)
{
    char a[256] = "1nteger_c";
    printf("Hi my name is %s", a);
}
```

```c
#include <stdio.h>

int main(void)
{
    char a[256] = "1nteger_c";
    printf("Hi my name is %s");
}
```

Format String Attack

```c
#include <stdio.h>
#include <unistd.h>
int main(void)
{
    char buf[256];
    read(0, buf, 100);
    printf(buf);
    return 0;
}
```

%n , $

- %n : 해당 주소를 받아서, “%n” 이전에 출력했던 문자의 개수를 해당 주소에 저장
- %n (4byte) , %hn (2byte) , %ln (8byte)

- $: 원하는 순서의 매개변수를 받아서 사용하게끔 해줌

- “%p%p%p%p%p%p%p%p%p%p” => “%10$p”

Format String Attack

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(void)
{
    print("system : %p", system);
    char buf[256];
    read(0, buf, 100);
    printf(buf);

    printf("/bin/sh");
    return 0;
}
```

Format String Attack

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
void shell()
{
    system("/bin/sh");
}
int main(void)
{
    char buf[256];
    read(0, buf, 100);
    printf(buf);
    read(0, buf, 100);
    printf(buf);

    printf("Bye!");
    return 0;
}
```

sprintf, snprintf

- 출력하는 데이터를 모두 buffer에 저장

- printf와 비슷하게 Format String Attack 가능