Segurança da Computação

Matheus Venturyne Xavier Ferreira

Universidade Federal de Itajubá

21 de Outubro de 2015

Web Attackes

- SQL Injection
- CSRF (Cross-site request forgery)
- XSS (Cross-site scripting))

Command Injection

- Executar código arbitrário no servidor

```
$in = $_GET['exp'];
Eval('$ans = ' . $in . ';');
```

http://sote.com/calc.php?exp=" 10 ; system('rm *.*')"

SQL Injection

- Exemplo PHP
- O que acontece se recripient é uma string maliciosa

```
$recipient = $_POST[‘recipient’];
$sql = "SELECT PersonID FROM Person WHERE
                                Username='$recipient'";
$rs = $db->executeQuery($sql);
```

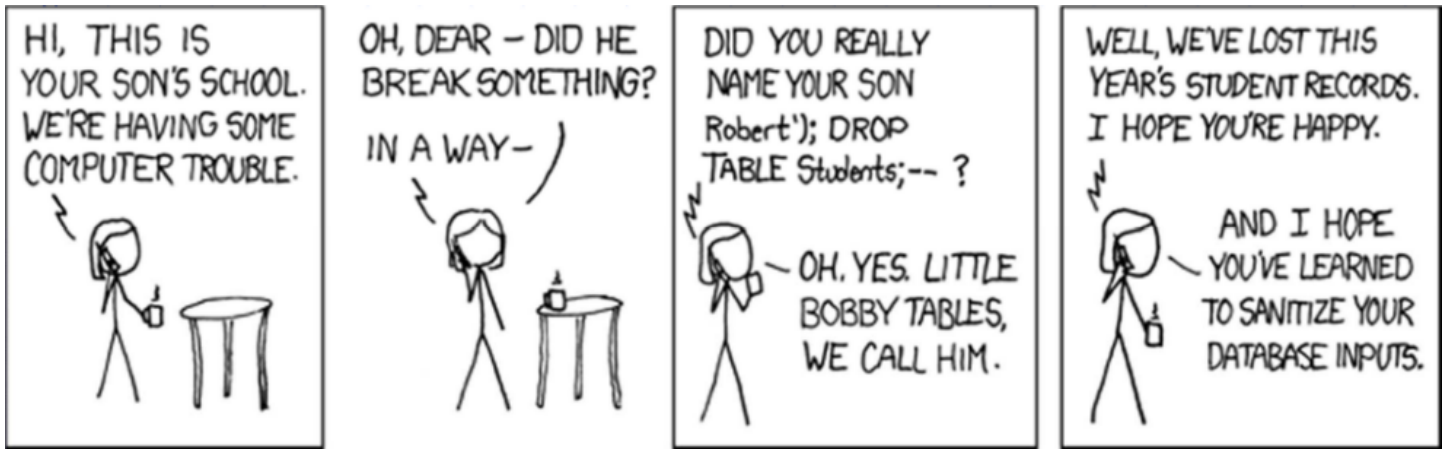
SQL Injection

```
Set ok = execute( "SELECT * FROM Users
    WHERE user=' " & form("user") & " '
    AND pwd=' " & form("pwd") & " `" );
If not ok.EOF
    login success
Else fail;
```

Suponha que o usuário entre

```
user = " '; DROP TABLE Users -- "
Ok = execute ( SELECT …
    WHERE user= '"; DROP TABLE Users … )
```

SQL Injection



Prevenindo SQL Injection

- Nunca construa commandos SQL
  - Use SQL preparado
  - Use ORM (Object-Oriented Mapping) framework

PHD: addslashes(“ ‘or 1 = 1 – “)

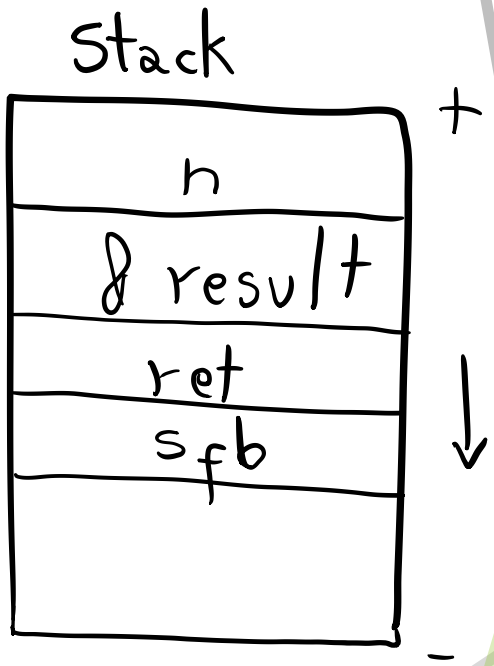
Outputs: “\’ or 1 = 1 – “

Cross Site Request Forgery

Ellipsis

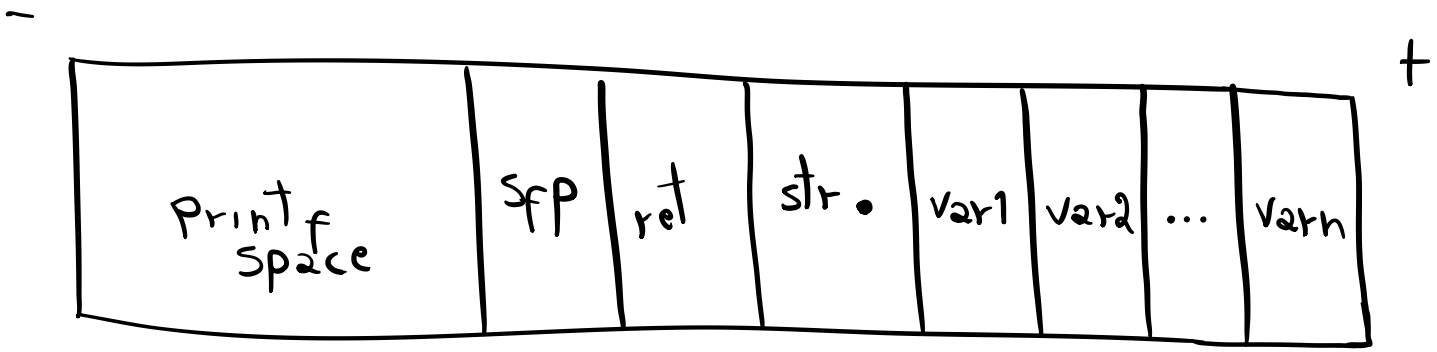
```
void func(int n, ...)
{
    va_list list;
    va_start(list, n);
    int arg = va_arg(list, int); // what am I
suppose to do here ._.
    arg = n + 7;
    va_end(list);
}

int main() {
    int result;
    func(10,&result);
    if (result == 17); // my goal
}
```



Printf

- Void printf(const char* str, …)



Printf

- A string passada para o printf possui a capacidade de ler e escrever informações na memória
- O que acontece se o usuário é permitido definer essa string?

```
string str;
cin >> str;
printf(str.c_str());
```

Printf - Parametros

- %n recebe um ponteiro da stack e escreve o número de bytes escritos até agora

parameter	*output*	*passed as*
%d	decimal (int)	value
%u	unsigned decimal (unsigned int)	value
%x	hexadecimal (unsigned int)	value
%s	string ((const) (unsigned) char *)	reference
%n	number of bytes written so far, (* int)	reference

Referências

- M. Zalewski: Browser Security Handbook, chapters 1 (basic concepts) and 2 (standard security features).
- D. Blazakis: Interpreter Exploitation
- Anonymous, “Once Upon a free()...,” Phrack 57 #0x09.
- Anonymous, “Bypassing PaX ASLR Protection,” Phrack 59 #0x09.