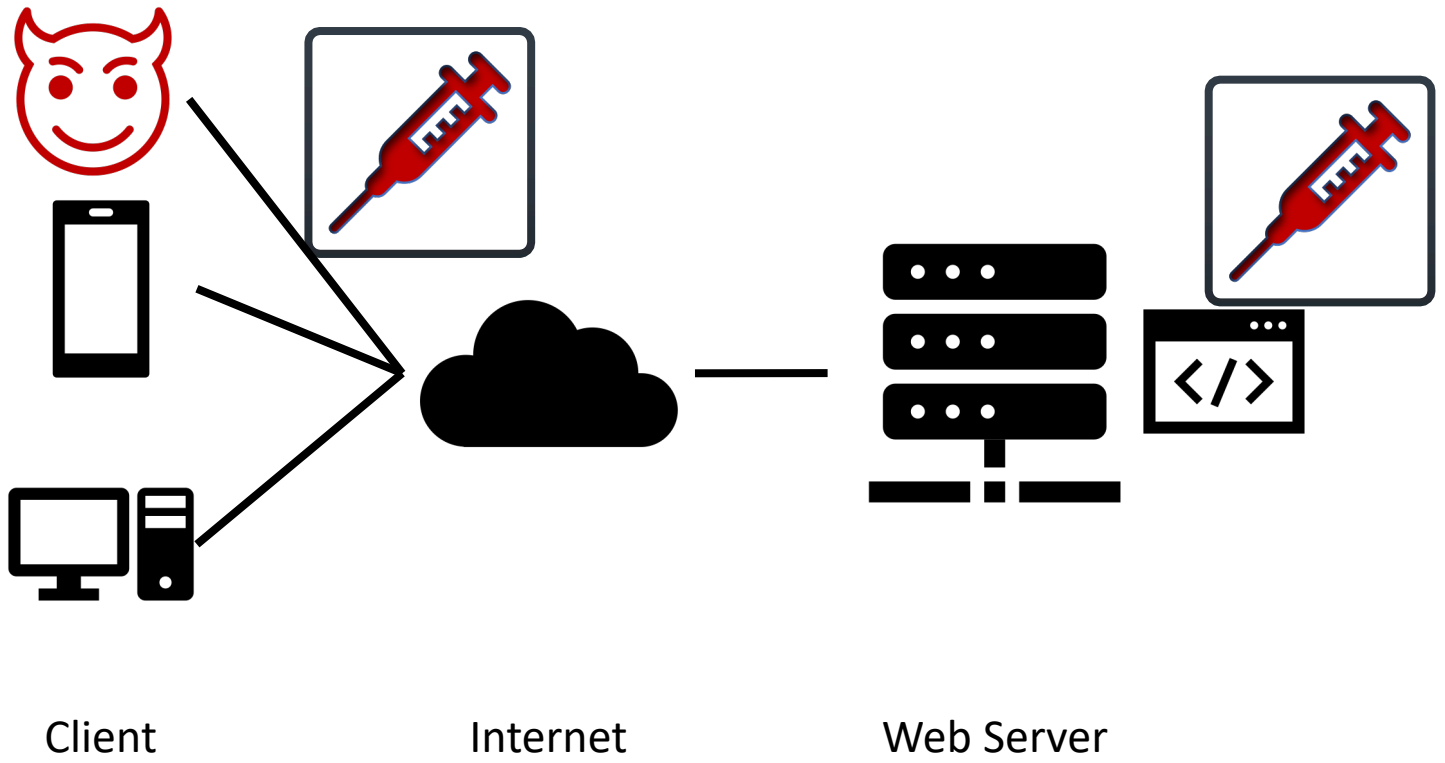
CS177

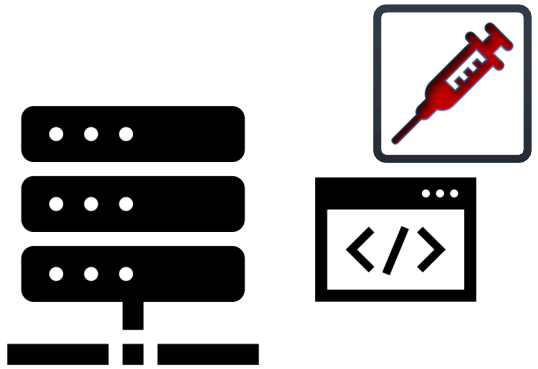
SQL & (OS) Command Injection

Hojjat Aghakhani, April 27, 2020

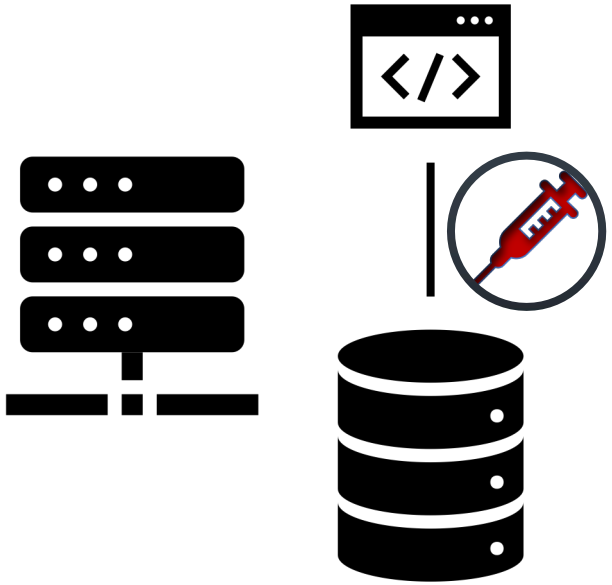
General Injection Attacks



General Injection Attacks



(OS) Command Injection



SQL Injection

OS Command Injection

- Web application code can drop into a shell to execute its command
  - In PHP by invoking system(), eval() or in Python, by invoking os.sytem()

- But what if these invocations use untrusted inputs with no validation?

OS Command Injection - Example

- What does the following url do?

```
cmd = "echo {}".format(form.getvalue('name'))
os.system(cmd)
```

- http://foo.com/echo.php?name=foo
- What about this one?
- http://foo.com/echo.php?name=foo; cat /etc/passwd

Linux command-line injection syntactical techniques

- Semicolons
  - cd etc; cat passwd
- Backticks
  - `ls`
- Pipes
  - ls | nc -l 8080

http://web.cecs.pdx.edu/~wuchang/courses/cs495/Slides/03a_A1_CommandCodeInjection.pdf

OS Command Injection - Example

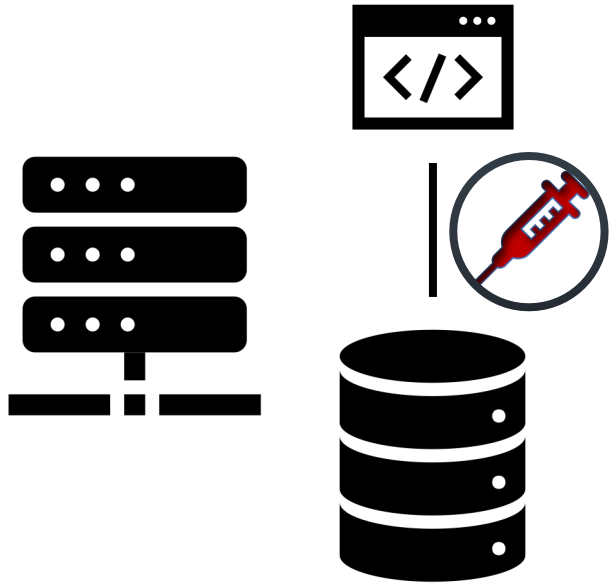
```
cmd = "echo {} > log.txt".format(form.getvalue('name'))
res = os.system(cmd)
```

- What if the attacker does not see the output?
  - For example: http://foo.com/echo.php?name=foo; cat /etc/passwd
- Assume application serves static resources from the filesystem location /var/www/static
- What about this?
  - http://foo.com/echo.php?name=foo& cat /etc/passwd > /var/www/static/passwd.txt
  - Then go to http:/foo.com/passwd.txt

How to prevent OS command injection attacks?

- The most effective way is never calling out to OS commands!
- If it is unavoidable to call out to OS commands, sanitize well and hope you are safe!

SQL Injection Attacks



SQL Injection Attacks

- A code injection technique targeting data-driven applications
- Malicious SQL statements/queries are executed by crafting unexpected input fields
- Goals:
  - Bypassing authentication
  - Extracting data
  - Modifying, adding or deleting data

Real World Examples

3 Indicted in Theft of 130 Million Card Numbers

The computers of Hannaford Brothers, a supermarket chain, were infiltrated.
Joel Page/Associated Press

Hacker breached 60+ unis, govt agencies via SQL injection

A hacker tied to the November 2016 penetration of the US Election Assistance Commission and subsequent database sale has successfully targeted 60+ government agencies and universities by leveraging the same attack method: SQL injection.

How Does SQL Injection work?

```
username = form.getvalue("username")
query = "select * from student where username='" + username + "'"
rows = db.execute(query)
print(rows)
```

- What does the following url do?
  - http://foo.com/echo.php?username=foo
- What about this one?
  - http://foo.com/echo.php?username=foo’ or 1=1
- Syntax error! There is a ` at the end
- What about this one?
  - http://foo.com/echo.php?username=foo’ or 1=1 --
- Yes! But what if they remove -- as a bad input?
  - http://foo.com/echo.php?username=foo’ or ‘1’=‘1

How Does SQL Injection work?

```
username = form.getvalue("username")
query = "select * from student where username='" + username + "'"
rows = db.execute(query)
print(rows[0])
```

- What if the server returns only the first row?
- It’s still an unauthorized access, but what if we want more?
- There are workarounds to bypass this limitation?
  - E.g., using Limit in your query:
  - http://foo.com/echo.php?username=foo’ or 1=1 limit 2, 2

Second-Order Attacks

- Not all SQL attacks are first order
- The attacker injects to a trusted database, a malicious entry
- An attack can be executed in a large scale by legitimate users

Defense Against SQL Injection

- Avoid constructing SQL queries by user inputs
- Comprehensive data sanitization
- Use a web application firewall (e.g., ModSecurity)
- Limit database privileges by context

https://www.slideshare.net/AnoopT3/sql-injection-69621267

Question?