

net square
secure.automate.innovate

Top Ten Web Attacks

Saumil Shah

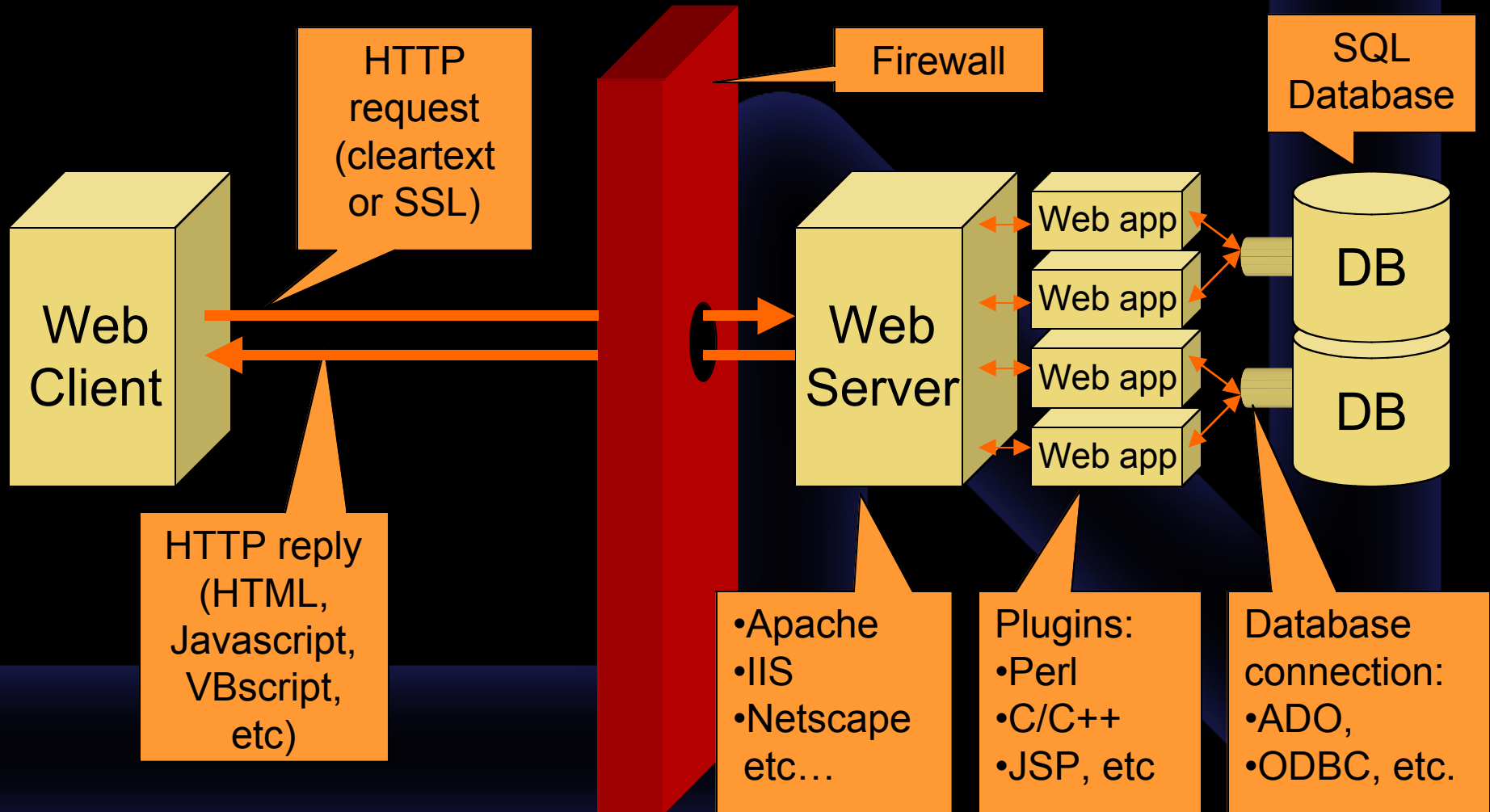
Net-Square

BlackHat Asia 2002, Singapore

Today's battleground – the Web

- Web sites and web applications rapidly growing.
- Complex business applications are now delivered over the web (HTTP).
- Increased “web hacking” activity.
- Worms on the web.
- How much damage can be done?
- Firewalls?

Typical Web Application set-up

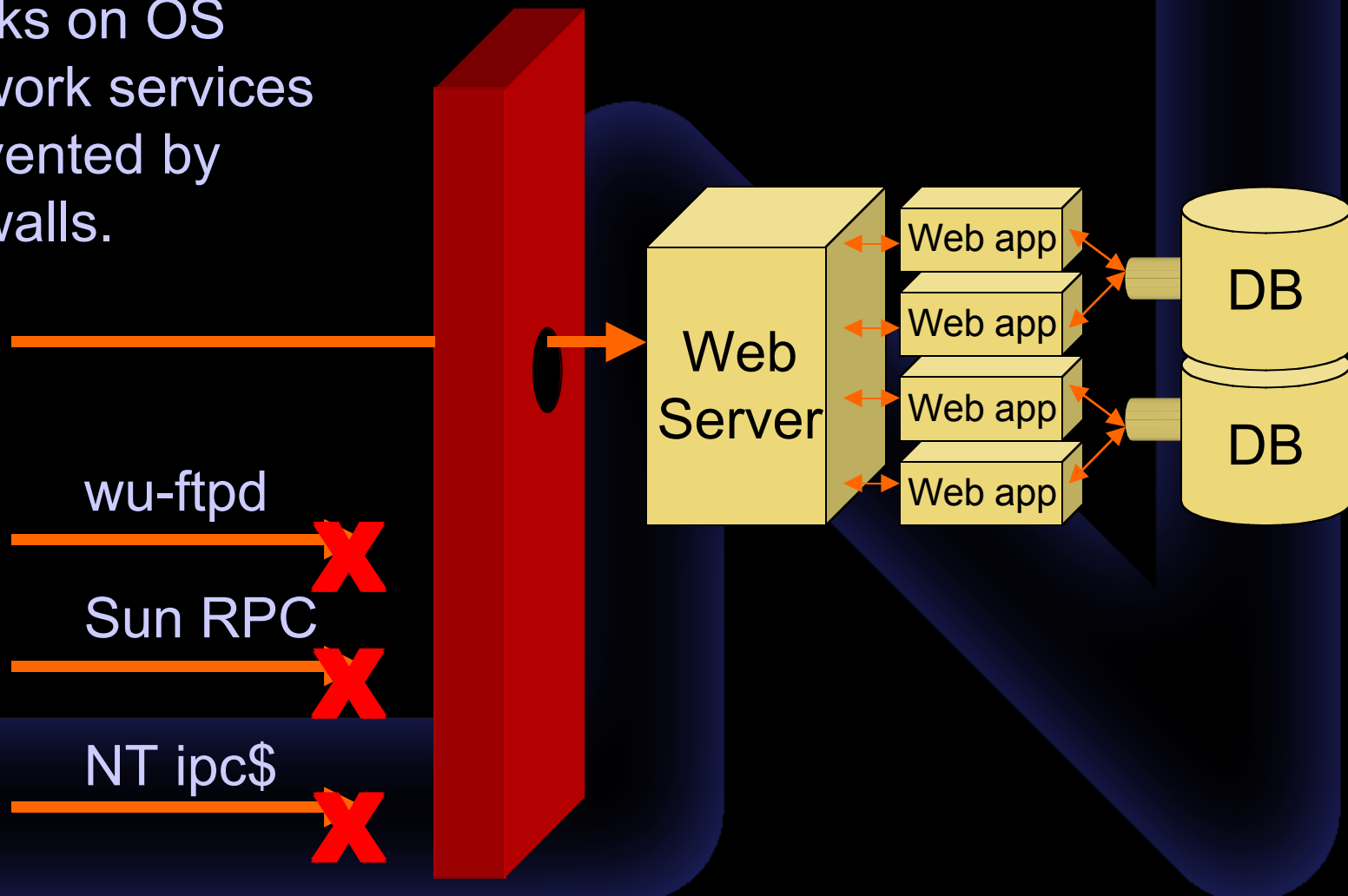


Traditional Hacking...Limitations

- Modern network architectures are getting more robust and secure.
- Firewalls being used in almost all network roll-outs.
- OS vendors learning from past mistakes (?) and coming out with patches rapidly.
- Increased maturity in coding practices.

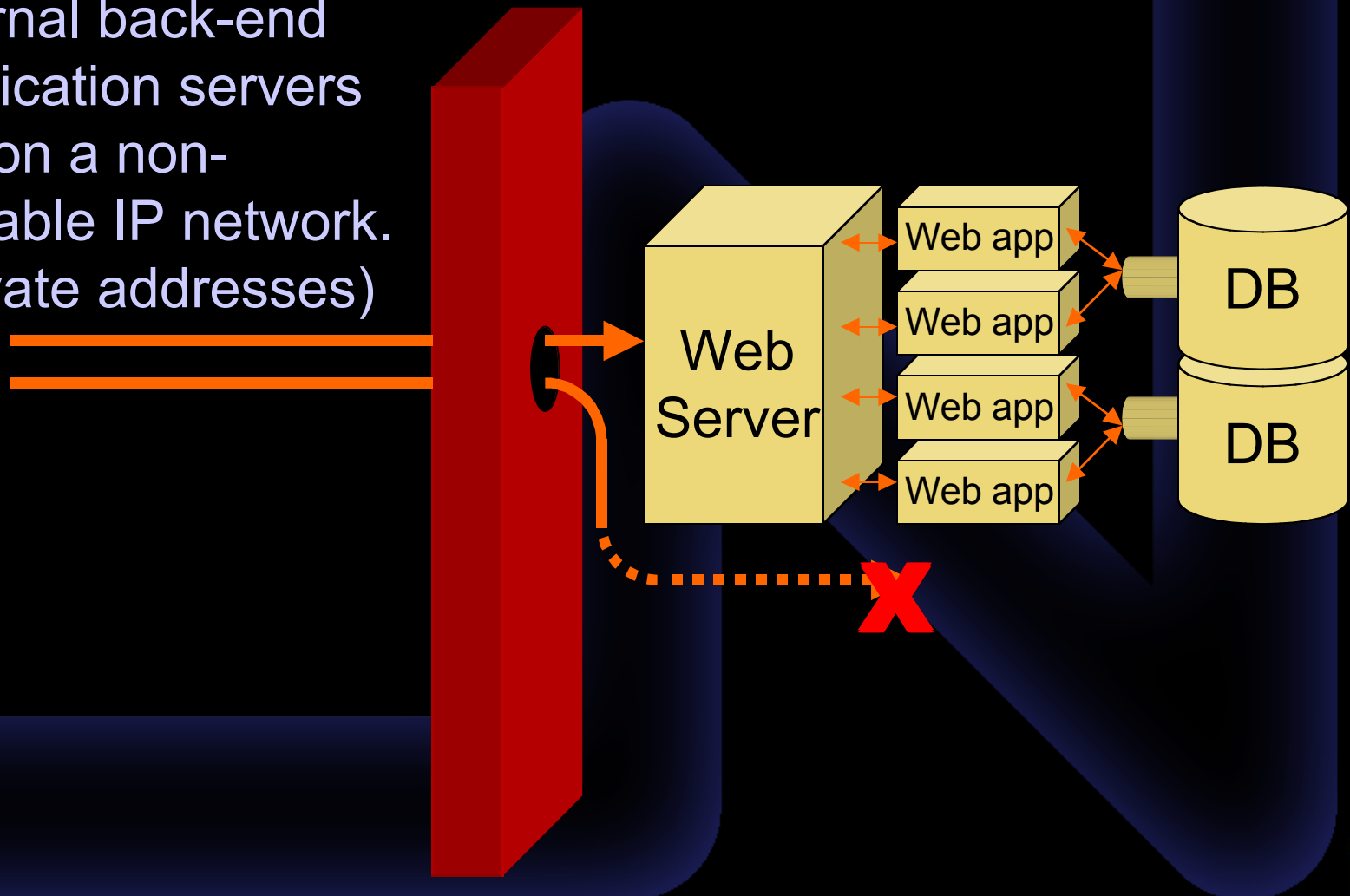
Utility of Firewalls

- Hacks on OS network services prevented by firewalls.



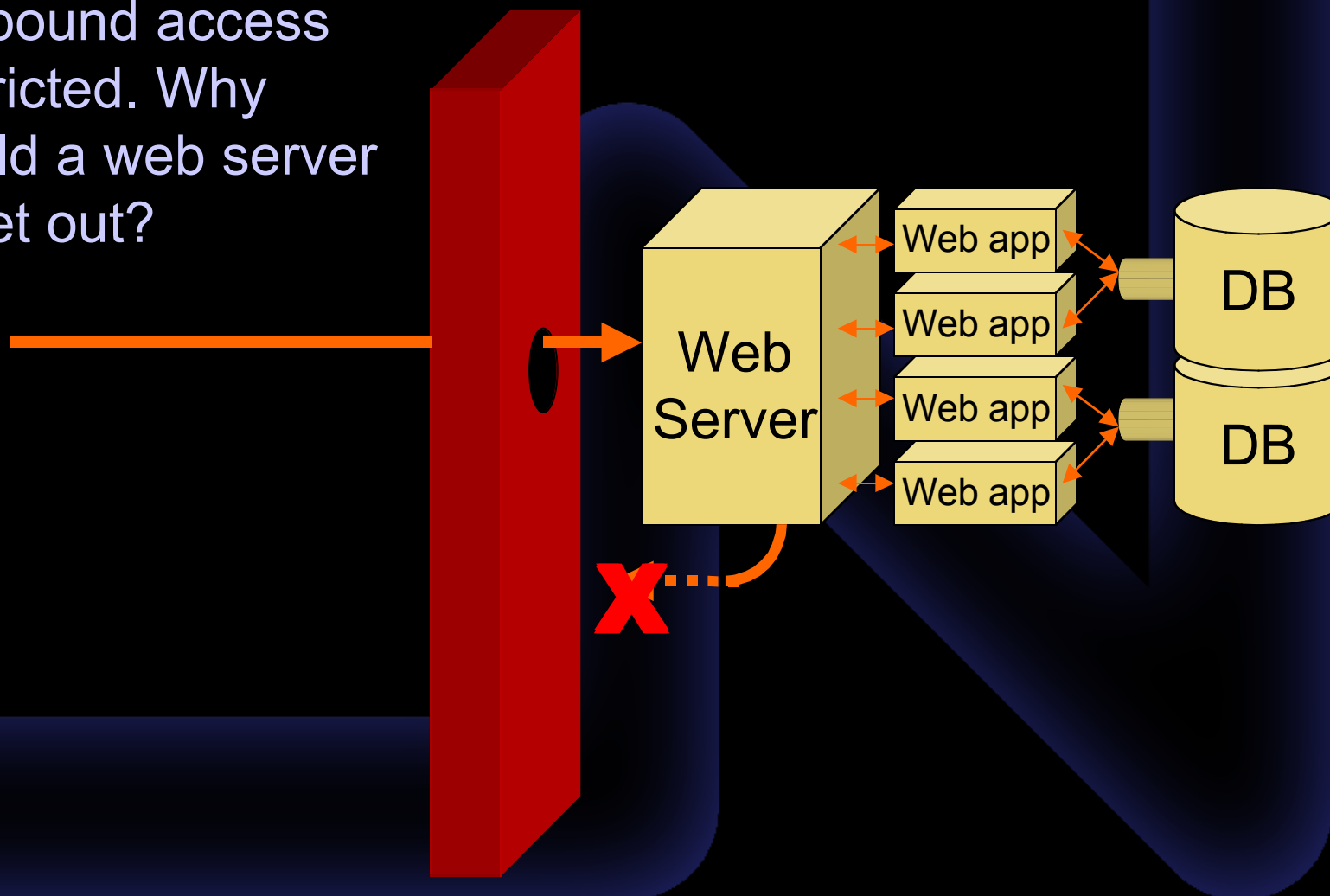
Utility of Firewalls

- Internal back-end application servers are on a non-routable IP network. (private addresses)



Utility of Firewalls

- Outbound access restricted. Why would a web server telnet out?



Futility of Firewalls

- E-commerce / Web hacking is unfettered.
- Web traffic is the most commonly allowed of protocols through Internet firewalls.
- Why fight the wall when you've got an open door?
- HTTP is perceived as "friendly" traffic.
- Content/Application based attacks are still perceived as rare.

The Web Hacker's Toolbox

Essentially, all a web hacker needs is ...

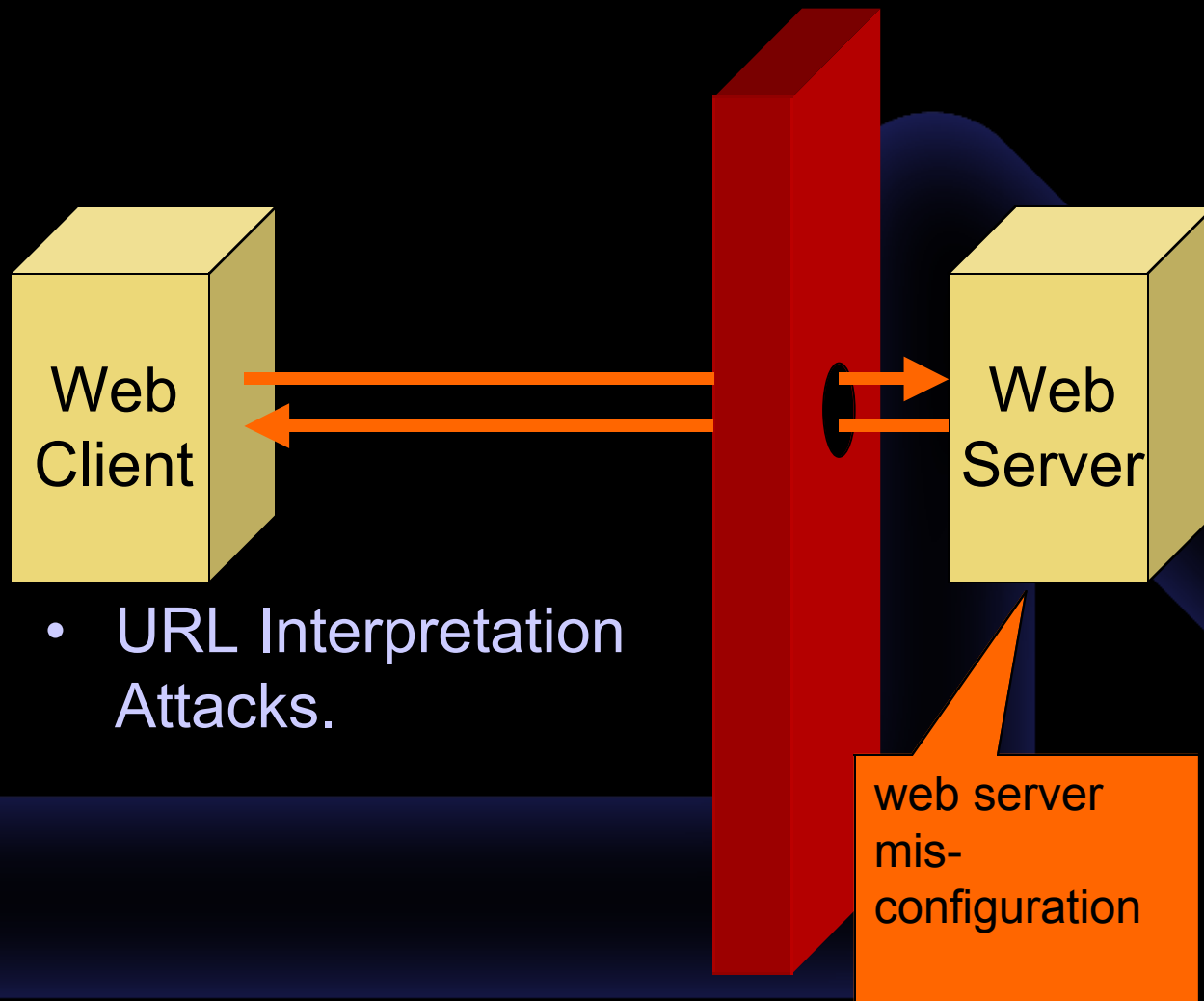
- a web browser,
- an Internet connection,
- ... and a clear mind.

Classifying Web Hacks

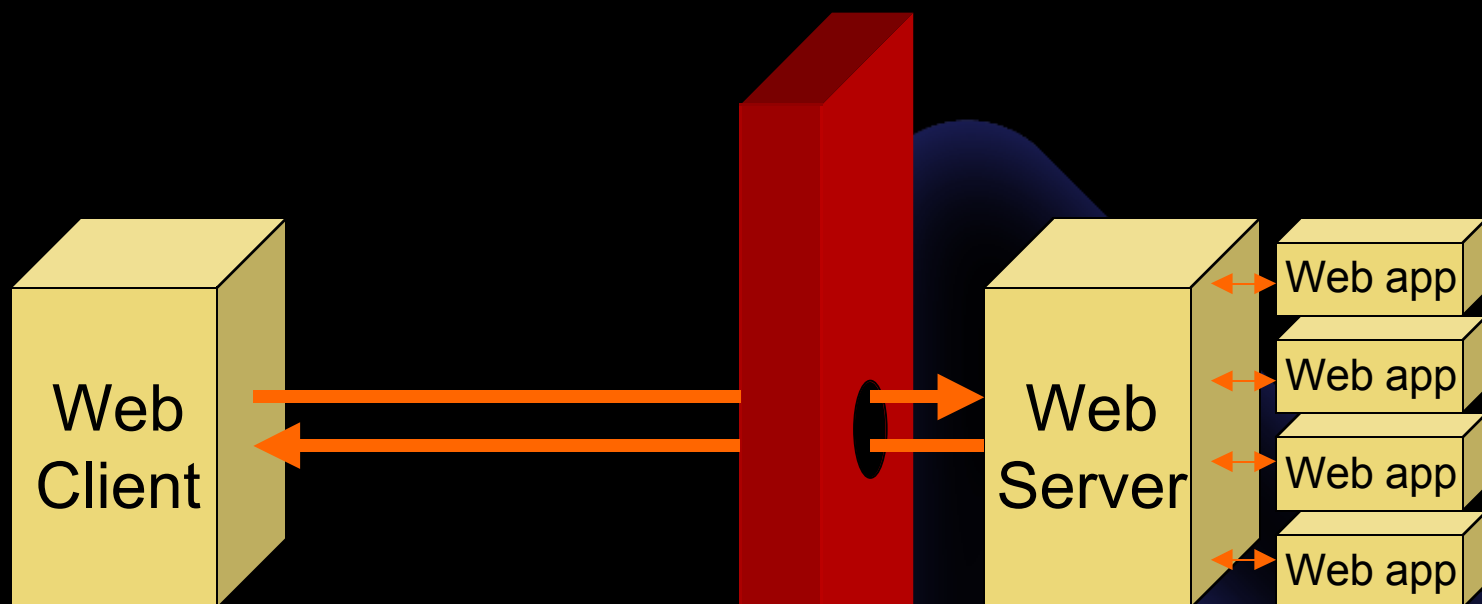
Web Hacks fall under the following categories:

- URL Interpretation attacks
- Input Validation attacks
- SQL Injection attacks
- Impersonation attacks
- Buffer Overflow attacks

Firewalls cannot prevent...



Firewalls cannot prevent...

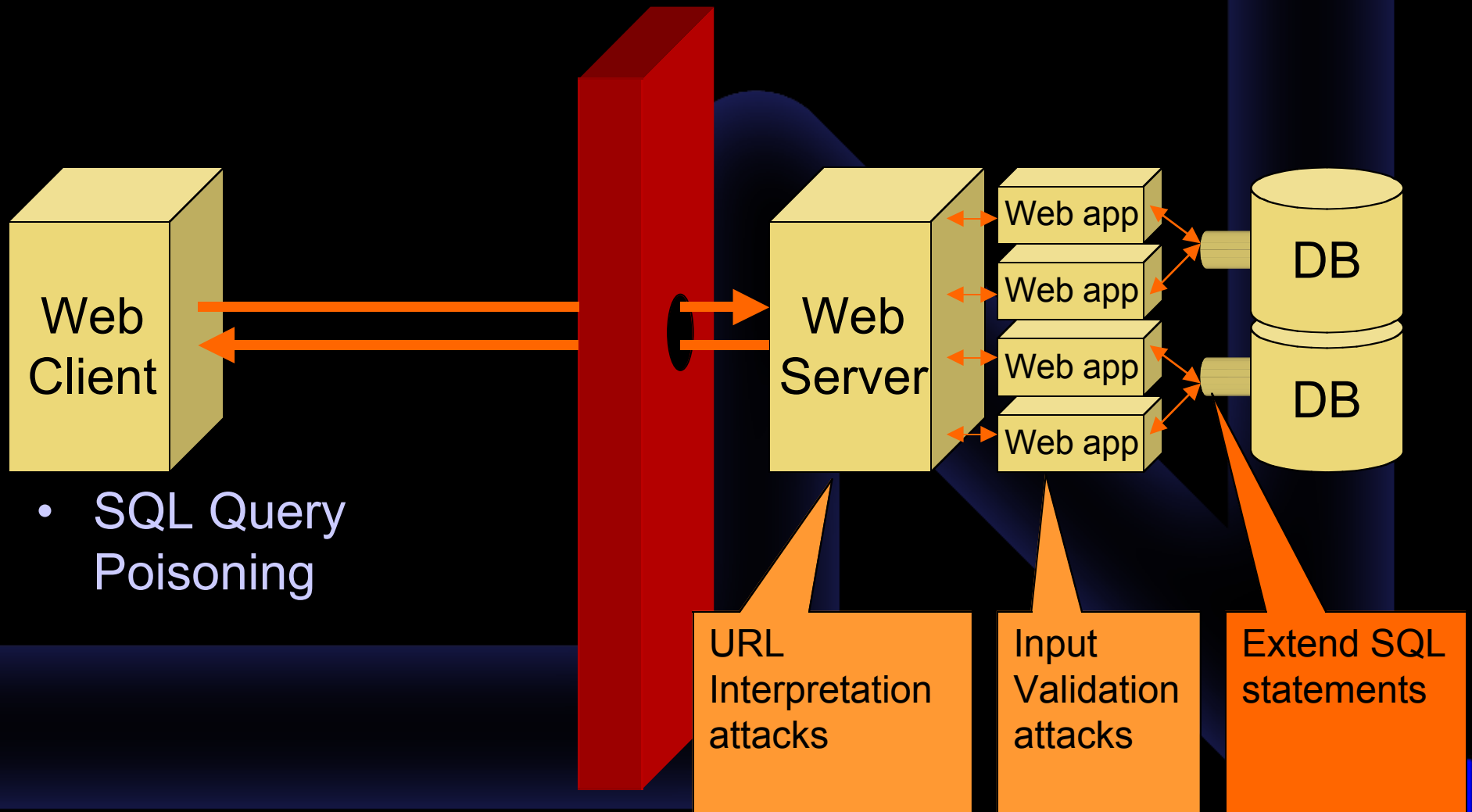


- Input Validation attacks.

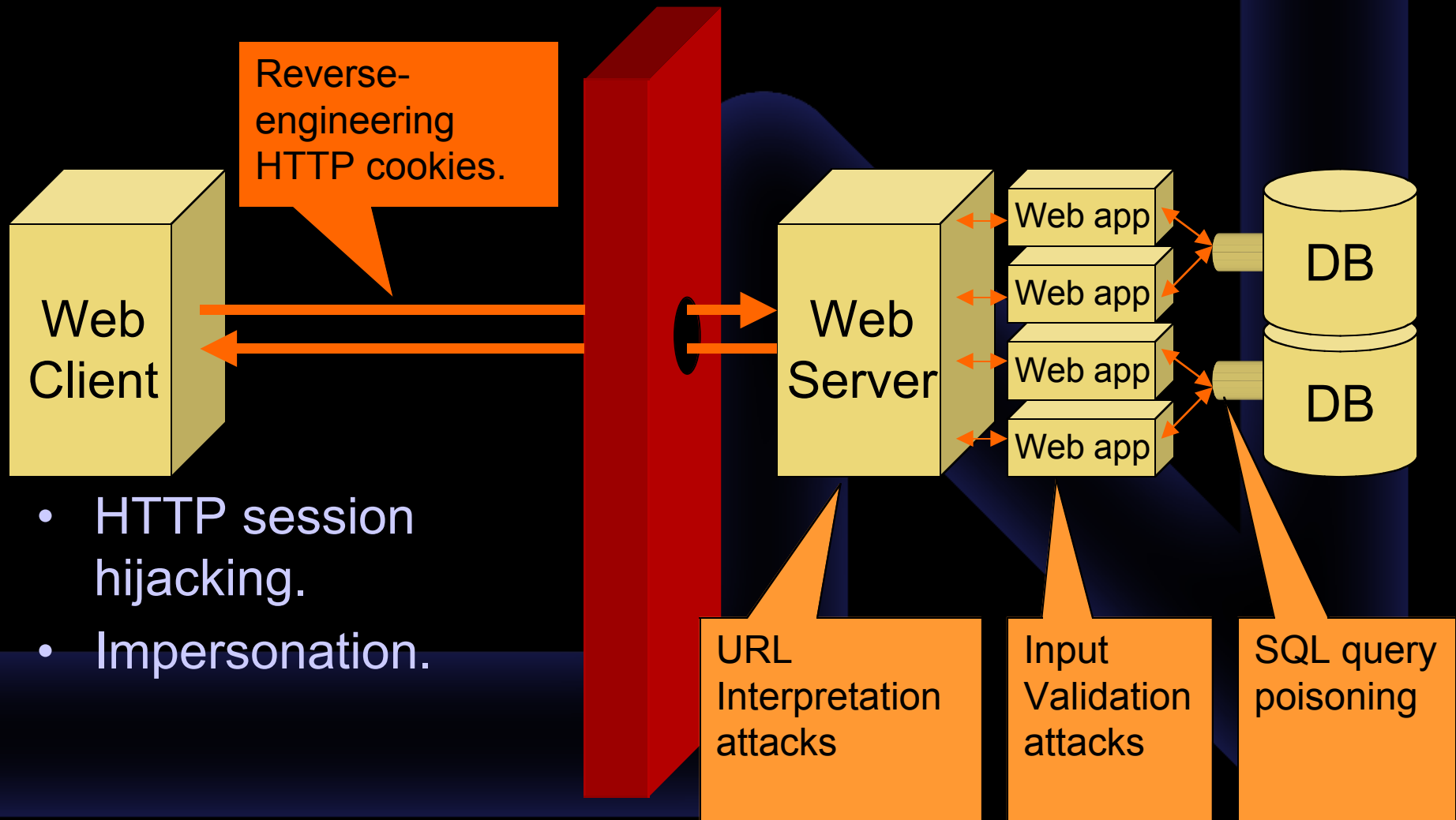
URL
Interpretation
attacks

poor
checking
of user
inputs

Firewalls cannot prevent...



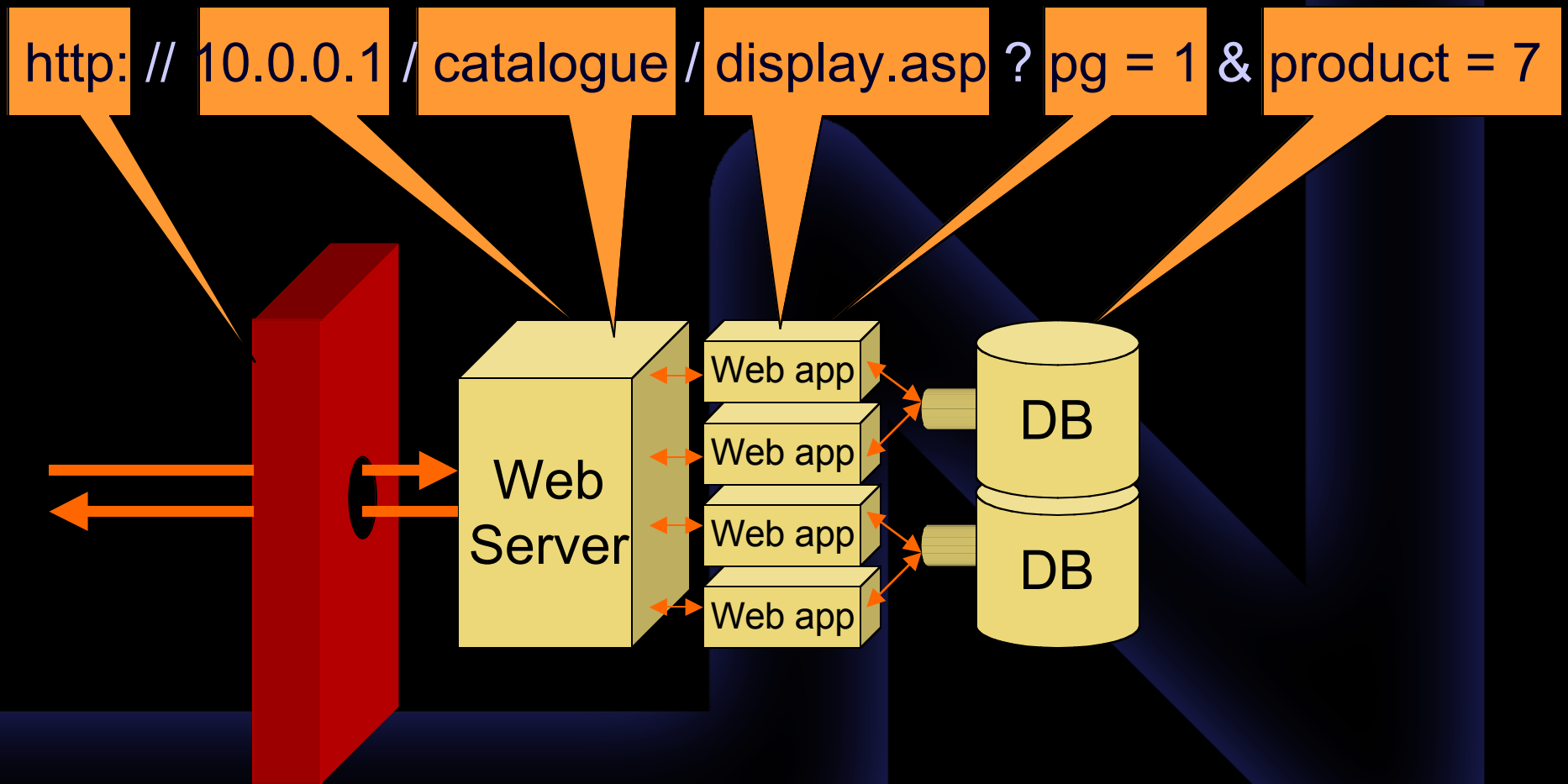
Firewalls cannot prevent...



Why is Web Hacking so deadly?

- Ports 80 and 443 are usually allowed through firewalls.
- A single URL works its way into many components.
- And in most cases, the only defense is “secure coding”.

The URL as a cruise missile



Web Hacks - net effects

Web Hacks cause three types of effects:

- Extra information disclosure. (paths, etc.)
- Source code and arbitrary file content disclosure.
- Extra data disclosure (e.g. return all rows)
- Arbitrary command execution.

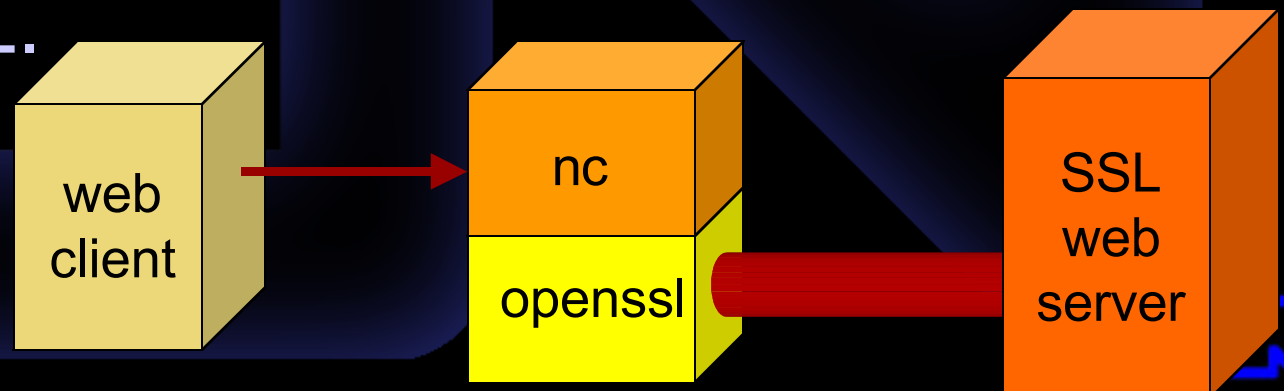
The Web Hacker's Toolbox

Some desired accessories would be ...

- a port scanner,
- netcat,
- vulnerability checker (e.g. whisker),
- OpenSSL, ... etc.

Hacking over SSL

- SSL Myth: “Strong 128 bit crypto stops hackers dead in their tracks”
- Using netcat and OpenSSL, it is possible to create a simple two-line SSL Proxy!
- Listen on port 80 on a host and redirect requests to port 443 on a remote host through SSL.



The Top 10 Web Hacking Techniques

1. URL Misinterpretation
2. Directory Browsing
3. Retrieving “non-web” Files
4. Reverse Proxying
5. Java Decompilation

The Top 10 Web Hacking Techniques

6. Source Code Disclosure
7. Input Validation
8. SQL Query Poisoning
9. Session Hijacking
10. Buffer Overflows

1. URL Misinterpretation

- The web server fails to parse the URL properly.
- e.g. the Unicode / Superfluous decode attack.
- Mismatched resource mappings in the configuration.
- e.g. `+.htr`, `.JSP`, Java remote command execution, etc.

1. URL Misinterpretation

Countermeasures:

- Usually require a vendor supplied fix.
- Thorough inspection of the web server configuration and bindings.

2. Directory Browsing

- Ability to retrieve complete directory listing within directories on the web server.
- Usually happens when the default document is missing.
- Not-so-strict Web server configuration.

2. Directory Browsing

Countermeasures:

- Web server configuration lock-down.
- Disable serving of directory listings.
- Sometimes the error may require a vendor supplied fix.

3. Retrieving “non-web” Files

- “Non-web” files can be:
 - Archive files (.zip, .tar.gz, etc)
 - Backup files (.bak, ~, etc)
 - Header / Include files (.inc, .asa, etc)
 - Text files (readme.txt, etc)
- Can be retrieved with some guess work.
- e.g. if there is a directory called /reports/, look for “reports.zip”.

3. Retrieving “non-web” Files

Countermeasures:

- Eliminate careless presence of such files.
- Disable serving certain file types by creating a resource mapping.
- Strict change control measures.

4. Reverse Proxying

- Web proxy servers may work both ways!
- Typically meant to allow users from within a network to access external web sites.
- May end up proxying HTTP requests from the outside world to the internal network.
- e.g. Compaq Insight Manager
- Usually happens when the front end web server proxies requests to back end app servers.

4. Reverse Proxying

Countermeasures:

- Check the web server proxy configuration thoroughly.
- Be careful when creating URL mappings to internal servers.

5. Java Decompilation

- Java Bytecode can be decompiled quite effectively.
- May disclose sensitive information such as passwords, application paths, etc.
- May also disclose application logic – such as generation of session IDs, encryption, etc.
- Java Archive files (.jar files) may contain files other than bytecode, such as configuration files.

5. Java Decompilation

Countermeasures:

- Java bytecode obfuscation.
- Elimination of sensitive configuration information within bytecode.
- Elimination of unnecessary files within .jar files.

6. Source Code Disclosure

- Ability to retrieve application files in an unparsed manner.
- Attackers can recover the source code of the web application itself.
- The code can then be used to find further loopholes / trophies.
- May be caused my many ways:
 - Misconfiguration or vendor errors
 - Poor application design, etc.

6. Source Code Disclosure

Countermeasures:

- Vendor supplied fixes.
- Locking down the web server configuration.
- Secure coding practices.

7. Input Validation

- Root cause of most web hacks.
- All inputs received should be validated:
 - data types
 - data ranges (e.g. -ve or fractional numbers)
 - buffer sizes and bounds
 - metacharacters
- Tampering with hidden fields.
- Bypassing client side checking (e.g. javascript).

7. Input Validation

Countermeasures:

- These are the worst to deal with!
- There is no other countermeasure but proper coding practices.

8. SQL Query Poisoning

- Parameters from the URL or input fields get used in SQL queries.
- An instance of Input Validation attacks.
- Data can be altered to extend the SQL query.
 - e.g. `http://server/query.asp?item=3+OR+1=1`
- Execution of stored procedures.
- May even lead to back-end database server compromise.

8. SQL Query Poisoning

Countermeasures:

- Again, no easy fix.
- Thorough source code review.
- Following the principle of least privilege for the database application.
- Elimination of unnecessary database users and stored procedures.

9. Session Hijacking

- HTTP is inherently a “stateless” protocol.
- Many web applications are stateful.
- Poor mechanisms of state tracking.
 - Hidden fields carrying a session ID
 - Client side cookies
 - ... with no server side session tracking.
- Reverse engineering of the session ID leads to access of other users’ data.

9. Session Hijacking

Countermeasures:

- Use server side session ID tracking.
- Match connections with time stamps, IP addresses, etc.
- Cryptographically generated session IDs.
 - hard to sequence.
- Use web application server session management APIs when possible.

10. Buffer Overflows

- Poor bounds checking.
- Web server HTTP requests.
 - e.g. ASP buffer overflow, .printer, etc.
- Application Input fields.
 - e.g. ColdFusion DoS, etc.
- Can cause:
 - Denial of service (crashing the app / service)
 - Remote command execution (shellcode)

10. Buffer Overflows

Countermeasures:

- Vendor supplied fixes.
- Bounds checking within applications.
- Source code reviews.
- Buffer overflow testing.

Hacking Web enabled Devices

- Network equipment, printers, etc. becoming “web enabled”.
- e.g. Cisco IOS HTTP hack, HP WebJetAdmin hack, etc.
- May leak sensitive information about a network.
- May allow proxying of web attacks.

Beating the IDS

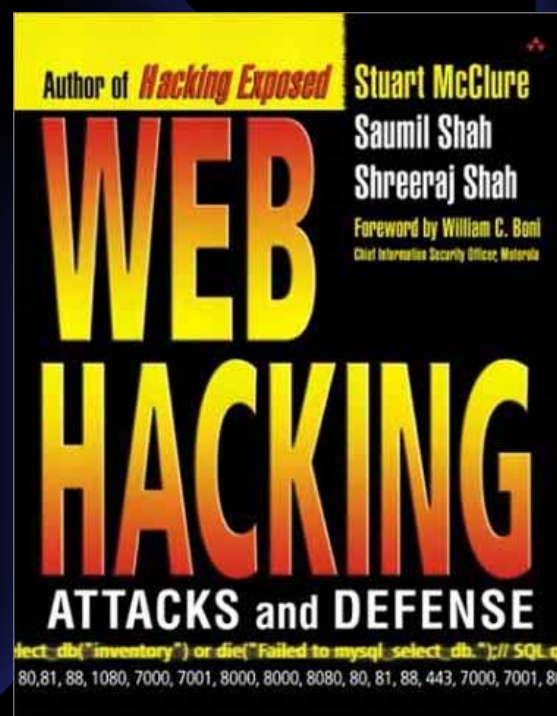
- “Secure Hacking” – hacking over SSL.
- Many ways of writing the same URL.
 - Defeats signature based pattern matching.
- Spurious parameters.
- Intentionally generating false positives.

Closing Thoughts

- Far harder to secure web sites and web applications.
- Need to create a heightened levels of security awareness.
- Use of formal software engineering methods for developing web applications.
- Use of secure coding practices.
- Thorough application testing.

Closing Thoughts

- “There is no patch for carelessness”.
- Web Hacking: Attacks and Defense
Saumil Shah, Shreeraj Shah, Stuart McClure
Addison Wesley – 2002.





net square
secure.automate.innovate

Thank you!

saumil@net-square.com

<http://www.net-square.com/>

+91 98254 31192