Application Security

More important

By

Fat bloke plus trivia boy

12/7/02

www.loud-fat-bloke.co.uk

More important - why

- Your applications are now exposed to external forces over the internet some of which are malevolent
- Your internet applications are now more advanced than just html and have greater function -> more open to subversion
- More demanding development timescales

I-spy with my little eye the 7 layer OSI

Content attacks

Double-decode Hack

HTML print Hack

PHP Bugs

SSL/Apache

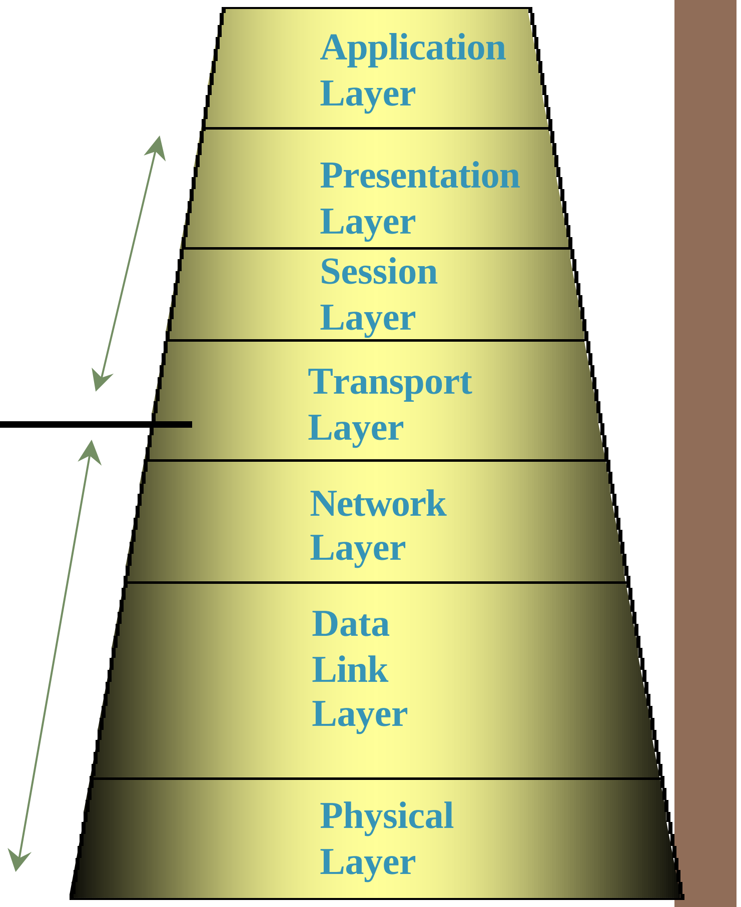
Context attacks

Fraggle

Evil ping

Port Scan

Sadmind



Application Layer

Presentation Layer

Session Layer

Transport Layer

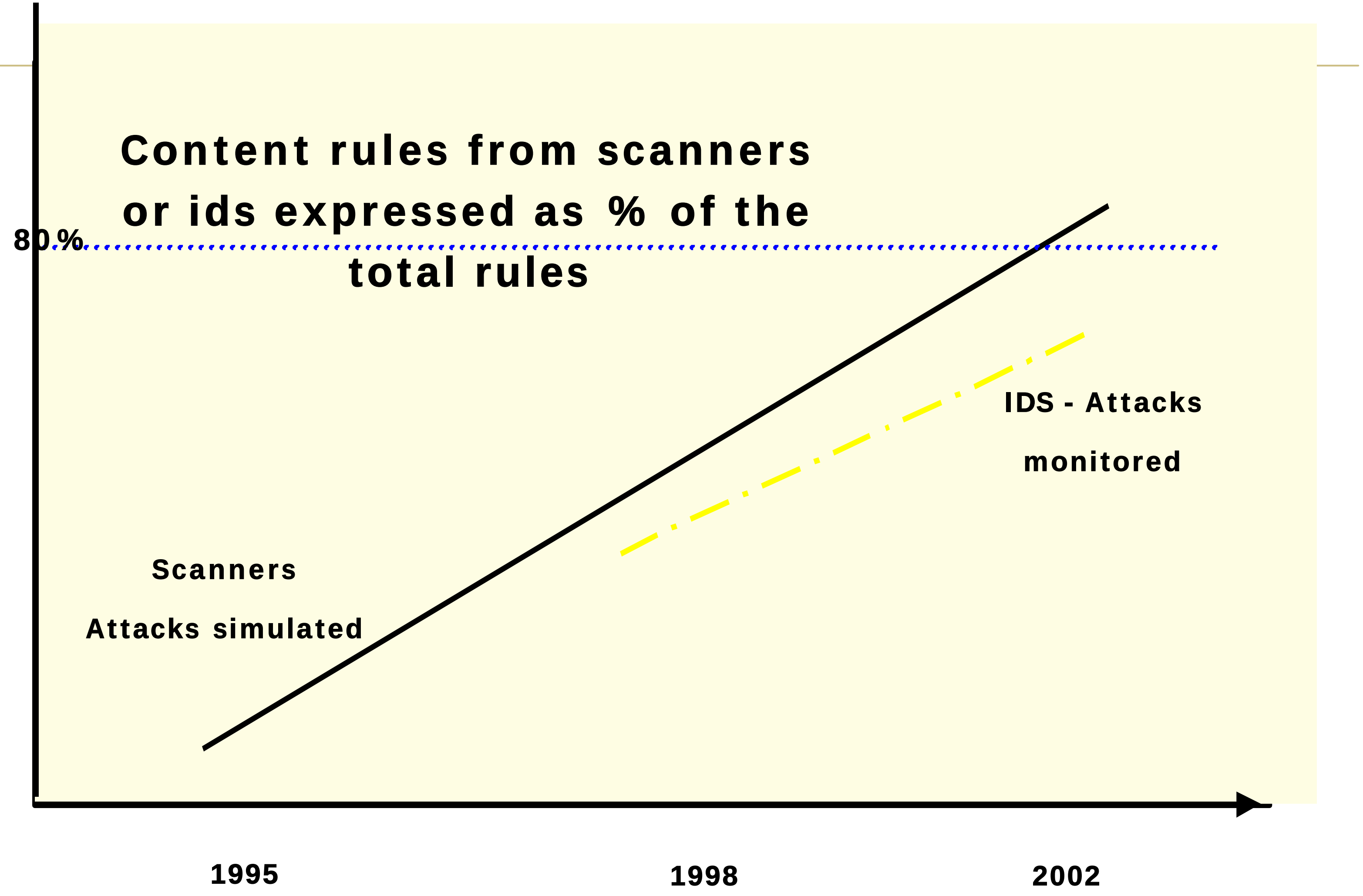
Network Layer

Data Link Layer

Physical Layer

7 layer osi model

Content attacks the Norm



12/7/02

The 7 layer OSI

Custom content attacks

Poor code in your bespoke Apps

General Content attacks

Double-decode Hack

HTML print Hack

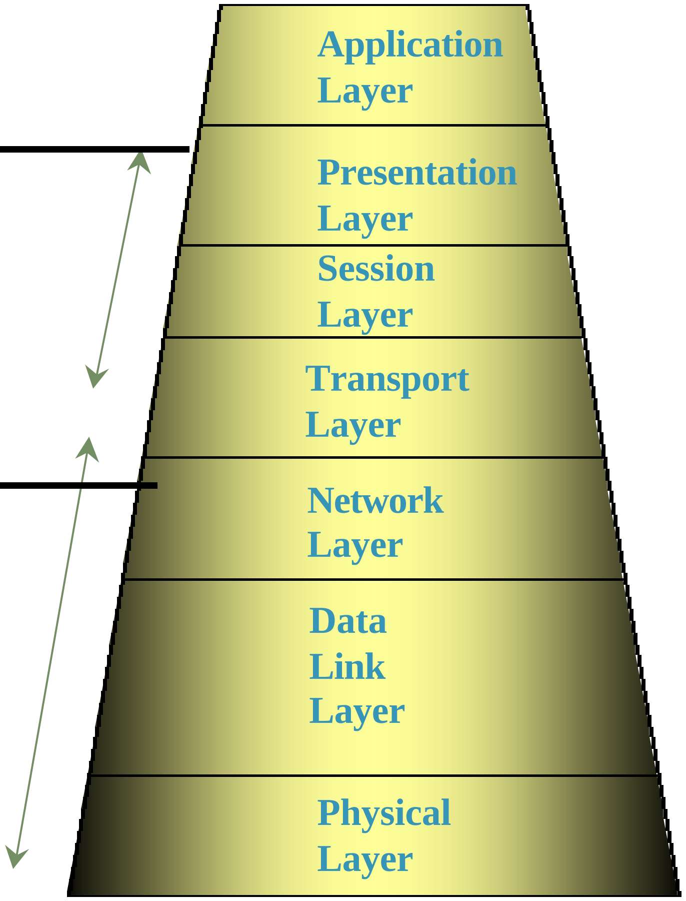
Context attacks

Fraggle

Evil ping

Port Scan

Sadmind



Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

7 layer osi model

Application Security

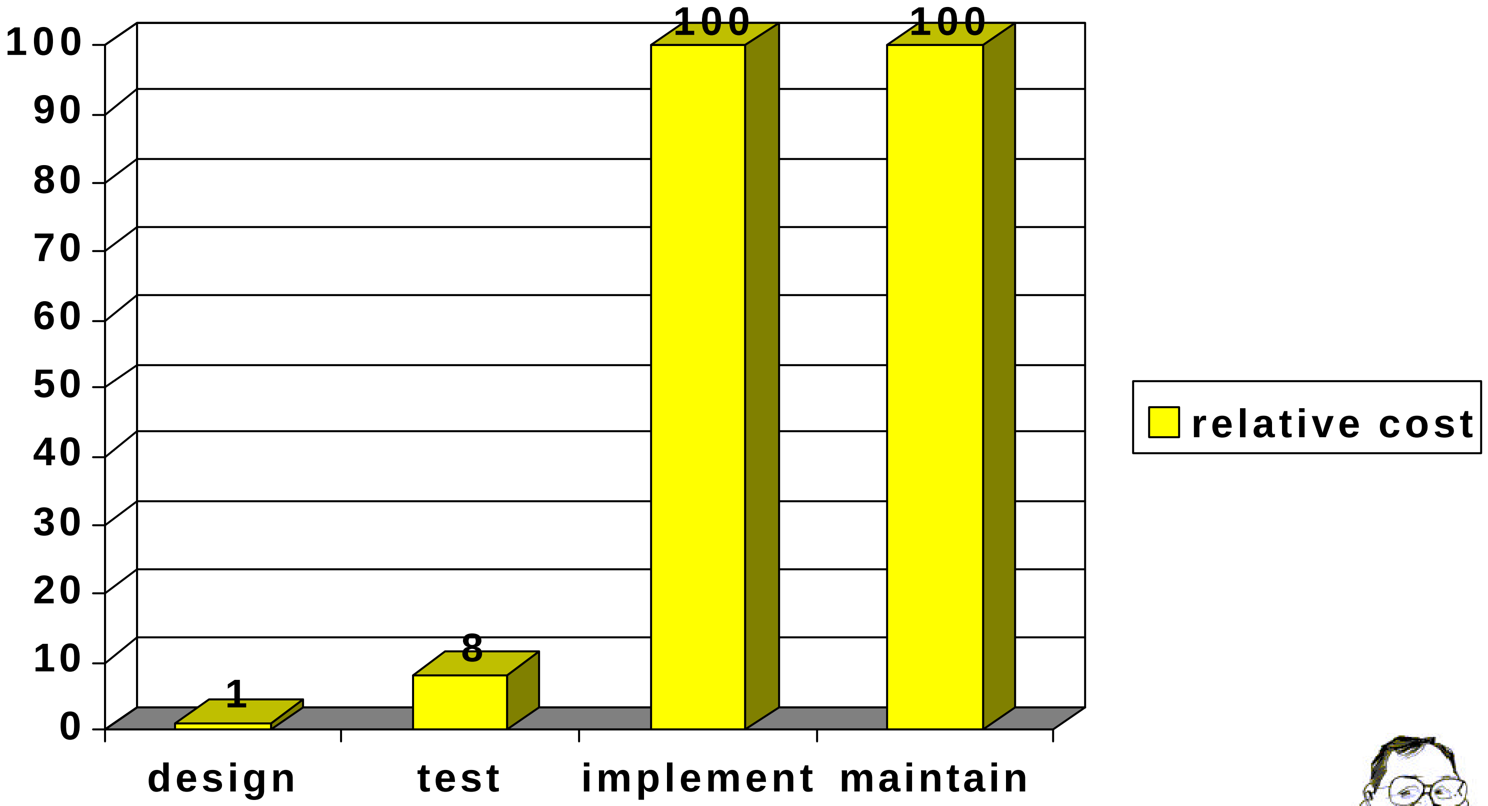
Cost of addressing

12/7/02

www.loud-fat-bloke.co.uk

Relative cost of implementing

By each stage of the SDLC



12/7/02

www.loud-fat-bloke.co.uk

Lets do it cheaply

12/7/02

www.loud-fat-bloke.co.uk

How does this help us

Give me a fish

And I will eat for a day

Teach me to fish

And I will eat forever

Our approach

1. Define programmer security standards
   - We learnt that operating systems are implemented insecurely without security baselines
2. Implement formal security design reviews
3. *Sitting with nelly* with the programmers and testers during UAT
4. Security Application pentesting at implementation

- ***This approach saves money***

www.loud-fat-bloke.co.uk

TOP TIPS

Web server configuration

This overlaps with infrastructure testing, however, web server and application configuration is often a major weak point. Common problems include, existing sample or help pages, unpatched or unhardened systems and unprotected administration pages.

F Ensure web servers are fully patched and all unused components are removed. If you use a web-configurator option on port 80 or 443, make sure that it is restricted on the web server to internal addresses or inhibited by a cvp rule on the firewall.

www.loud-fat-bloke.co.uk

Site obfuscation

An attacker will always examine the application for information. This means they will look for HTTP headers, server banners, test pages, browsable directories or unusual errors. Without the information that such surveys reveal, it is significantly harder to attack the site.

F Ensure error messages and “href”s don’t disclose too much info (this may help site portability anyway)

Authentication

With a transaction-based system where usernames, passwords and other SPIs are required, the authentication system is a common attack target. Login credentials should be strong enough to resist guessing and brute forcing and transmission of credentials should be encrypted with SSL. Authentication management is often dealt with using session identifiers and cookies.

F Ensure that session IDs cannot be predicted or guessed, and that cookies are secure, and are session based rather than persistent. Also ensure the authentication system is fully integrated through-out the system rather than just implemented on a login screen, which might not be the only entry point to the system. Lastly, store these SPIs securely.

Inter process communication and Session Management I

Data has to be passed between different parts of an application to maintain state, often these can be manipulated by attackers to gain extended privileged. Common exposures include:

- URLs – by changing directory or file names to gain access to other parts of the application
- URL query strings – by altering parameters that follow the "?" in GET requests
- HTTP Headers – by altering timeout and referrer settings
- POST data – by intercepting and altering posted parameters
- Hidden tags – by altering the parameters sent by a page but unseen by the user
- Cookies – by reverse engineering and altering session ID content

www.loud-fat-bloke.co.uk

Inter process communication and Session Management II

F Ensure Cookies are protected and are session based. Avoid URL based transmission of credentials as it is begging for trouble. Inputs should be adequately validated. Furthermore, most attacks against authorisation and state rely on altering information at the client side. The use of server side session tables will help prevent these attacks.

www.loud-fat-bloke.co.uk

Input Validation I

Attackers will attempt to alter data whenever a web client sends information to a server. The intention is to send data that the application is not expecting to cause a result or error. This can be as simple as sending very long strings to cause a Denial of Service effect or as sophisticated as inserting SQL commands or JavaScript to execute hostile code.

Buffer overflows are a common symptom of this error.

www.loud-fat-bloke.co.uk

Input Validation II

F Input validation should be performed on the server side as client side input validation, such as using JavaScript can usually be easily overcome. Ensure that data sent is strongly typed, that raw errors are not sent back to the user and that all server software is run with least privilege.

www.loud-fat-bloke.co.uk

Comments

Every trainee programmer gets told to comment their code – However, nearly every job we do uncovers security related information in code provided as a comment.

F Consider stripping comments from production servers

www.loud-fat-bloke.co.uk

Logs

Logs and temporary files often contain user details and sometimes passwords

F Consider storing them away from areas where they can be accessed by attackers I.e. so they are not browsable from the internet

System segregation

Perhaps a repeat of point 1. Many applications contain several sections – only one of which needs to be available to the internet.

F Ensure adequate planning and design takes place to prevent internal portions of an application being accessed externally.

www.loud-fat-bloke.co.uk