



# Why Web Applications are making a hackers life easy.

Presented by Jon Grew BT SBS



# Acknowledgements

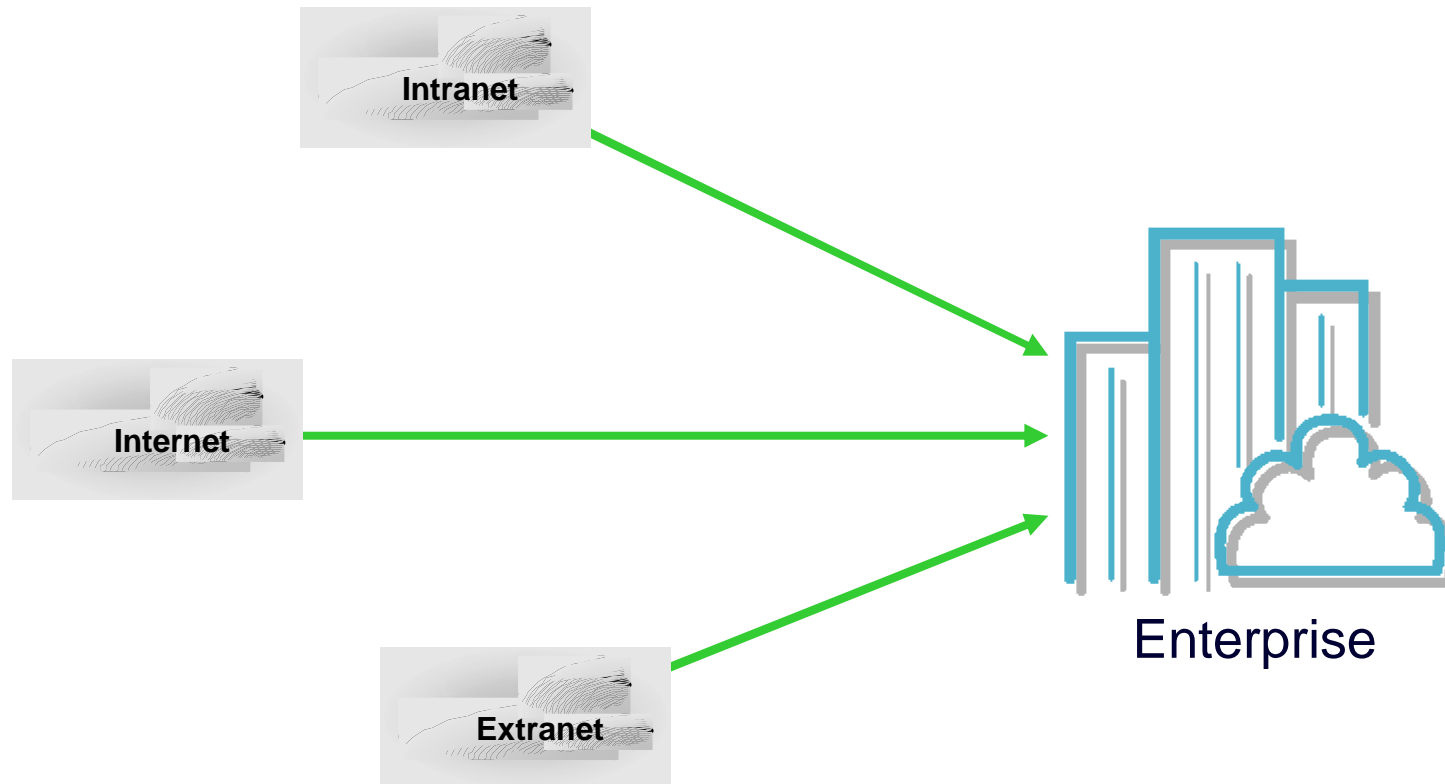
- Ed Barlow      Technical Director EMEA



Ed sends his apologies. The following presentation is based on the talk he was meant to give today.

Ed originally gave a version of this presentation at Infosec this year.

# Attacking the Enterprise



# Is it a real risk ?

Unprotected Websites are attacked an average of 2,000 times a week, a new study has revealed.

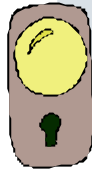
Security firm PanSec International and Internet service provider PSINet Europe set up two fake banking sites and monitored the number of times they were attacked over an eight-week period. One site was protected with a standard firewall – the other was left unprotected.

The firewall prevented 90 per cent of attacks, but the protected site was still attacked more than 200 times a week.

**The unprotected site was attacked a total of 19,128 times over the eight-week period – more than once every five minutes.**

# Source Code has Errors.

One Security Vulnerability

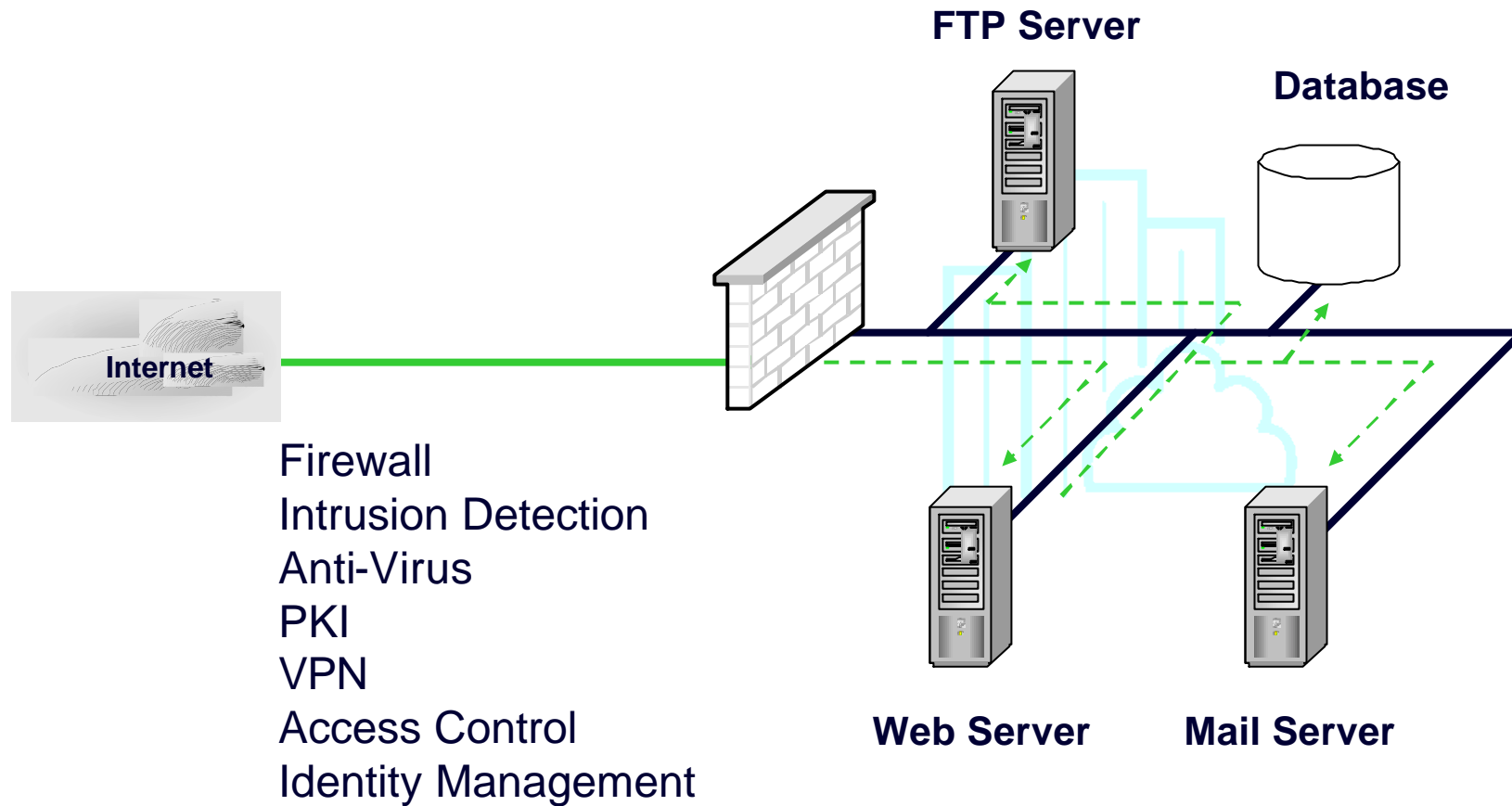


1,500 lines of code

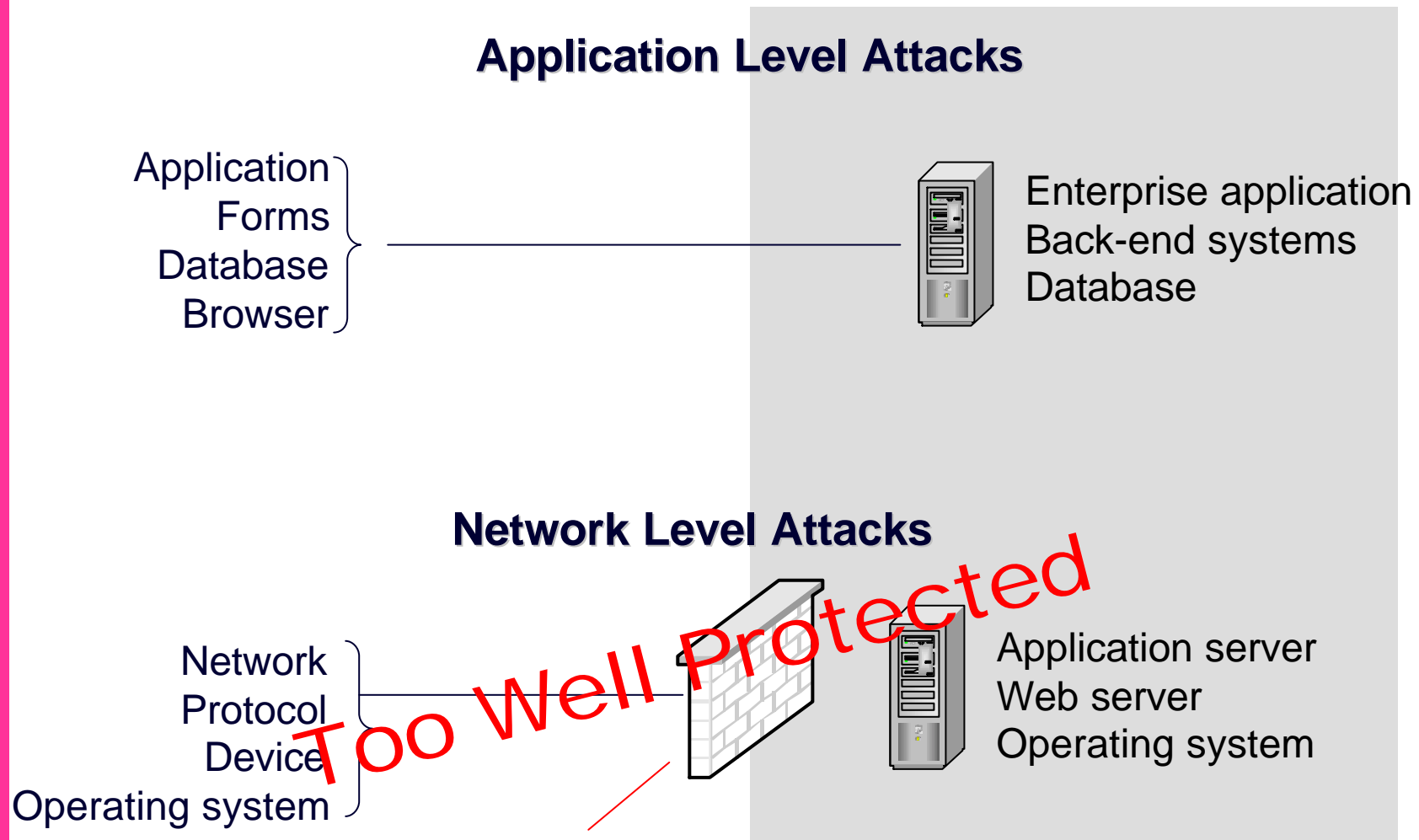
*IBM Watson Research Lab*

|              | <b>Lines of code</b> |
|--------------|----------------------|
| Windows 2000 | 35 million           |
| Linux        | Over 30 million      |

# Protecting the Enterprise



# Attackers Move With The Times



# Attackers Move With The Times

## Application Level Attacks

Application  
Forms  
Database  
Browser

SQL Injection  
Parameter Tampering  
Cookie Poisoning  
Vulnerability Patterns  
HTTP Methods Exploitation  
Application Language Mismatch  
3rd Party Mis-configuration  
SOAP & Web Services Message Exploitation  
Files Upload  
Protocol Piggyback  
Buffer Overflow Attack  
Data Encoding

Network  
Protocol  
Device

Operating system

Application  
d systems

on server  
ver

Operating system

*Too Well Protected*

# SQL Injection Attack

## Threat description

Information from Web requests is not validated before being used by a Web application to access data in an SQL database. Intruders can use these flaws to get the database to perform requests that were not intended. E.g bypassing password controls

## Example

The application expects a user to submit a parameter which will be used as part of a select statement to validate a login and password.

Expected value: `http://www.myWeb.com/find.asp?Login=Admin&password=123456`

The intruder sent: `http://www.myWeb.com/find.asp? Login=Admin&password= 123456 ' OR 1=1`

## Impact

Forces the application to change the SQL submitted to the database and bypasses logic controls as `1=1` is all ways true .

The user is logged in no matter what the password.

# Example

## PetCo target of FTC Investigation

### SECURITYFOCUS NEWS

#### FTC investigates PetCo.com security hole

By **Kevin Poulsen**, SecurityFocus Dec 5 2003 5:08PM

Pet supply retailer PetCo disclosed this week that its security and privacy practices are the target of an investigation by the U.S. Federal Trade Commission (FTC), which is following up on an e-commerce security gaffe that left as many as 500,000 credit card numbers accessible from the Web earlier this year.

In October the FTC served PetCo with a "Civil Investigative Demand" seeking information and documents on how the company protects private customer information on the PetCo.com e-commerce site, PetCo revealed in its quarterly report Wednesday. "At the present time, the Company is unable to determine whether the FTC will initiate any enforcement action against the Company or the financial impact any such action might entail," the company wrote.

# Parameter Manipulation Attack

**Ranked as #1 threat in Web applications**

## Threat description

Information from Web requests is not validated before being used by a Web application. Intruders can use these flaws to get information from a database.

## Example (a real world example)

The application expects a user to submit a parameter value with a length of 8 bytes (numbers & characters)

Expected value: `http://www.myWeb.com/find.asp?ArticleID=34ABC67Y`

The intruder sent: `http://www.myWeb.com/find.asp?ArticleID=34`

## Impact

The intruder was able to get ALL document numbers starting with 34xxxxxx, which included confidential information

# Example

## Victoria's Secret Pursued by New York Attorney General

### **Victoria's Secret fined for Web security flaw**

---

Published October 23, 2003

Victoria's Secret has agreed to pay a \$50,000 fine to the state of New York while promising to improve computer security practices after a glitch on its Web site allowed viewers to browse other customers' online orders.

The lingerie stores' Columbus, Ohio -based parent company, Limited Brands, said it fixed the problem within days of being notified by a customer in November. New York Attorney General Eliot Spitzer announced the fine and settlement with Limited on Tuesday.

A glitch in a feature allowing customers to check their order status allowed them to randomly call up other orders, seeing details such as sizes, prices, customer names and addresses. The faulty site didn't reveal credit card numbers or allow visitors to search orders by name.

The company is notifying about 560 customers who were affected nationwide by mail, a spokesman said.

# Cross Site Scripting Attack

## Threat description

A script is embedded in a field contained in the URL. When executed the script adds, replaces or overlays fields onto the original page to capture information and post it to a rogue site.

## Example

The application expects a user to submit a parameter value, which it presents back on the returned page. By embedding HTML in the parameter, it is HTML which is embedded in the returning page.

Expected value: `http://www.myWeb.com/find.asp?SearchStr=Hacking Articles`

The intruder sent: `http://www.myWeb.com/find.asp?SearchStr=<script>alert("You Have Been Hacked")</script>`

## Impact

By getting a user to click a crafted link on a web page or email, this results in a change to the way the original pages is intended to behave.

# Example

## Plaxo plugs phishing vulnerability

[Munir Kotadia](#)

**ZDNet UK** March 16, 2004, 13:25 BST

**Plaxo has plugged a gaping security hole in its Web site that could have exposed its members' online address books.....**

.....a script added an additional layer over the username and password box. With this layer in place, if a user typed in their access details, the information would first be sent to the attacker's Web site and then to Plaxo to log the user in. Users would have had no idea their details had been taken.....

## Hotmail

[http://www.whitehatsec.com/labs/advisories/WH-Security\\_Advisory-08152001.html](http://www.whitehatsec.com/labs/advisories/WH-Security_Advisory-08152001.html)

## Many of the phishing scams

<http://www.antiphishing.org>

# Tools required

- A Web Browser

Unlike vulnerability assessment, no exploit code or Network scanners are required. This means that an attacker does not need to be able to write in C or wait for an exploit to be released.

# Problem Resolution

- Do nothing
- Re-engineer the application
- Other methods

# Do Nothing

- No immediate cost
  - Existing legislation being strengthened
    - Data protection
  - New regulation
    - Corporate governance
  - Loss of credibility
    - Damages value built in corporate & product branding
  - Financial penalties
    - PetCo; Victoria's Secret; ...
  - Current & future revenues due to attacks
  - Application level attacks are now well known
    - If attacks succeed and there were no precautions
      - Ignorance is no excuse for inaction

# Fix the Application

## New Applications

- Security reviews
- Incorporate extensive input checking



## Existing Applications

- Regular security reviews
- Update code to prevent new attacks

## Third Party Components

- Security patch & update programme

“Given the multitude of possible attack methods, any data from the user – even a simple HTTP request – should not be implicitly trusted”

# Development Considerations

## Input checking example: Character encoding

- Single quote character commonly used in SQL injection attacks
  - Can be encoded in 6 ways
  - Each is valid within SQL
  - Could also be valid user input
- "<SCRIPT>" alone represents a total of  $8^6$  possible combinations.

## Third Party Components

- Patch development delay
- Cost & time to install patches
- O/S & application vendors have security reviews in place...

## Existing Applications

Take offline to fix major problems  
Can only fix problems you know about  
Code change risk

90 percent of cyberattacks by 2005 will exploit security flaws for which a patch is available or a solution known

*Gartner*

# The Real Challenge - Complexity

## Web Technologies

|             |                     |
|-------------|---------------------|
| HTTP 1.1    | HTML 4.1            |
| XML 1.1     | Flash Objects       |
| SOAP        | Applets             |
| WebServices | Future Technologies |

## Web Environments

Single server – *simple*

Multi servers – *cluster*

Single server / Multi applications – *Virtual hosting*

Multi servers / Single application – *Large deployment*

Multi servers / Multi applications – *Farms*

High performance

High availability

## Web Application Attacks

SQL Injection

Parameters Tampering

SOAP & WebServices

Cookie Poisoning

Cross-Site Scripting

Known Vulnerabilities

HTTP Exploits

Default Configurations

Application Flow

Buffer Overflow

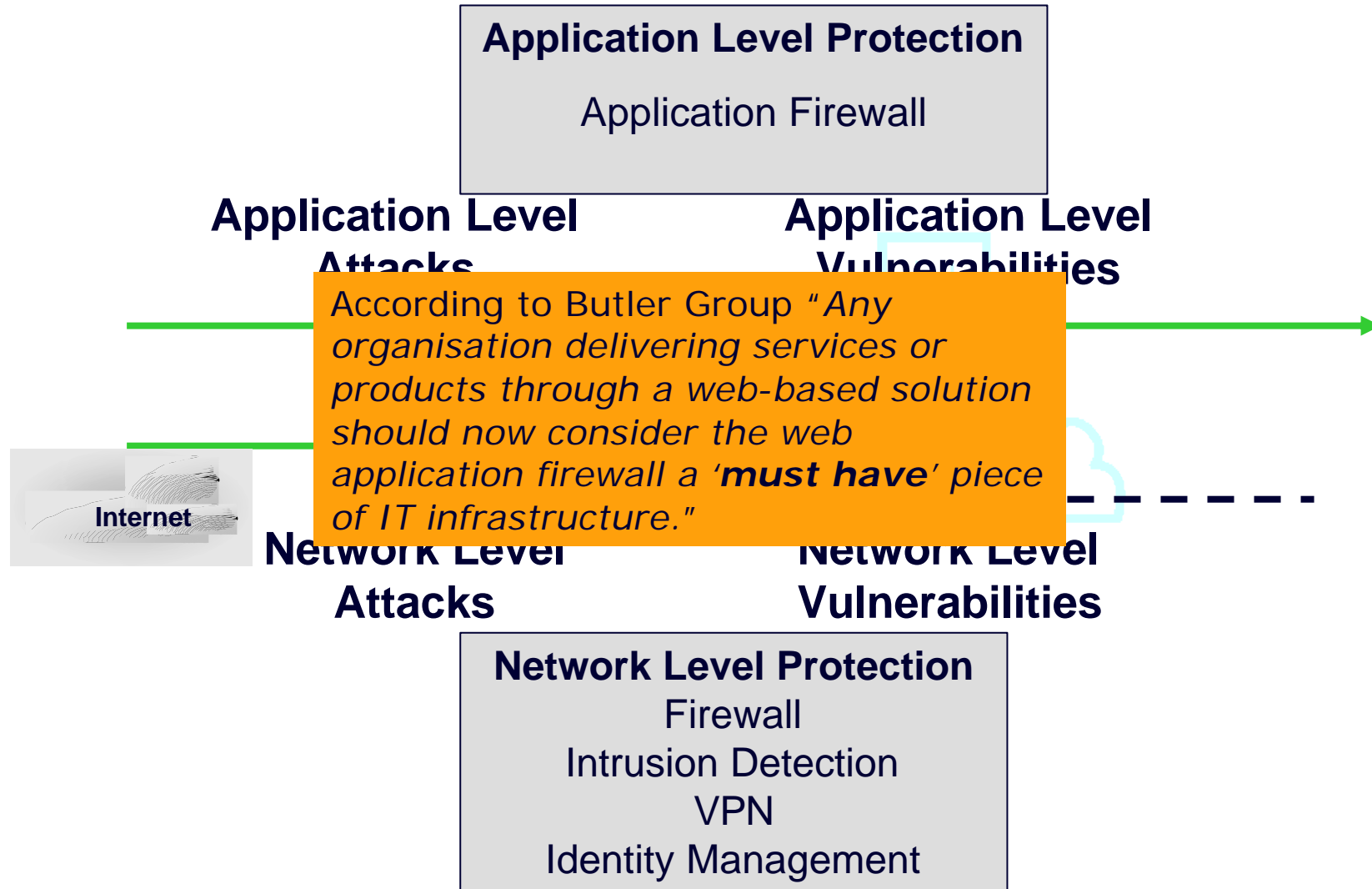
Files Upload

Data Encoding

Backdoor & Debug



# Alternatives



# Alternatives Application Firewall

- No application changes
  - Reduced cost & time to market
  - Protects against known and unknown attacks
    - No time offline waiting for patch
    - No exposure during patch development
  - Protects third party components
    - Minimise pressure to install patches
  - Relieves commercial pressures
    - No application downtime
    - No risky production system changes

# Resources

- Application scanners
  - Scando <http://www.kavado.com>
  - Appscan <http://www.sanctuminc.com>
  - Webinspect <http://www.spidynamics.com>
- Application Firewalls
  - Interdo <http://www.kavado.com>
  - Appshield <http://www.sanctuminc.com>
  - Codeseecker <http://www.owasp.org>

