



---

# **1000 application tests: what have we learned?**

---

Nick Ellsmore, CTO, **stratsec**

---

**stratHACK briefings, Canberra, August 2010**

# Background – Source Data

- Private sector clients
  - 200+ clients
  - 60% of ASX/S&P 20
  - 50% of ASX/S&P 50
- Public sector clients
  - Most Federal Government departments
  - Many State Government departments (esp. VIC/NSW/QLD)
- Our largest clients are in critical sectors:
  - Banking & finance
  - Utilities (energy, transport)
  - Defence
  - Federal and state government
  - ...and the IT companies who support these sectors

# Why Test?

- Things aren't always the way the design document says
- The fact someone can design something well doesn't necessarily mean that it will be built well
- Regulatory compliance requirements
- Due diligence
- Contractual requirements with partners or clients
- Internal policy or standards compliance requirements
- To get comfortable with what your systems are letting people do

# Why do problems still exist?

- Flaws in underlying technology
- Compressed application development schedules
  - Squeeze on security
- Exposure to the world
  - People are more creative than we often give them credit for
- Step-by-step guide for conducting an attack
  - Not always a technical challenge
- Interconnectivity of systems
  - Gateway to business critical systems
- Real incentives

# What have we noticed and seen?

- Evolution of development platforms
- Increased use of cryptography (not necessarily effectively)
- Complexity increasing
- Some problems getting attention (XSS/SQLi), others getting ignored (CSRF)
- Although web-app testing is popular, thick client applications are often worse (and get tested less)

# Platforms

- One step forward...



- Microsoft ASP.NET leading the way and drastically reducing many common high risk vulnerabilities in new platforms.
- Frameworks for other modern platforms are doing the same but with less co-ordination and market penetration.

- One (or more) steps back...



- Vendors with products built on older platforms are fighting to stay alive.
- Still building on old, insecure by default platforms such as ASP Classic, PHP, and Cold Fusion.
- These vendors need to invest in rebuilds of their products!

# Platforms

- One step forward...



- Better platforms = less work to do for end-customers and developers
- Common issues such as SQL injection and cross-site scripting are seeing a downturn.

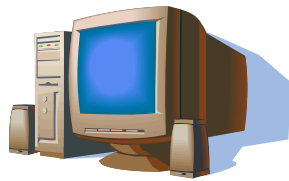
- One (or more) steps back...



- Organisations and developers are getting complacent
- Business logic style issues, such as authorisation bypass are seeing an upturn as frameworks cannot inherently prevent them.

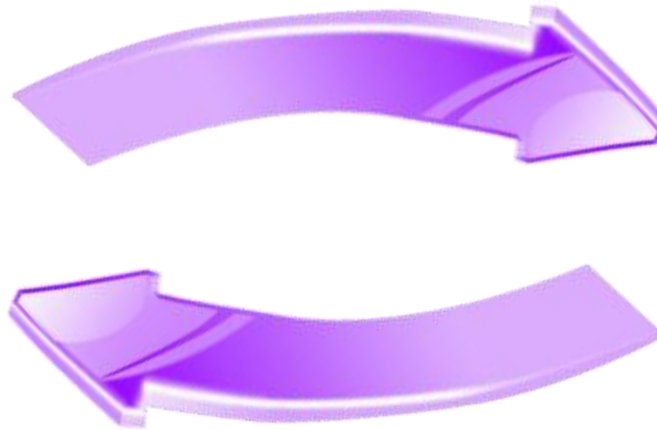
# Cryptography

- **Positive:**
  - Developers are catching on that cryptography can help solve many problems
  - Fewer people are complaining about performance impact
- **Negative:**
  - They are either solving the wrong problems or implementing it poorly
  - It's often being used to obfuscate poor security



Server

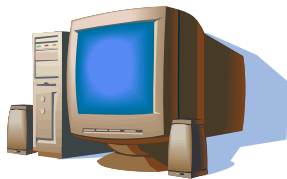
Parameter (Encrypted)



Parameter (Encrypted)

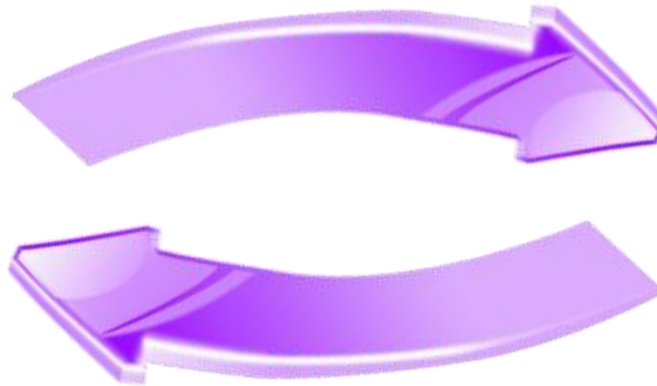
Reasoning: Avoid parameter tampering

Dangerous logic: If decryption works, it must be safe



**Server**

**Parameter (Encrypted)**



**Incorrectly Tampered  
Parameter (Encrypted)**

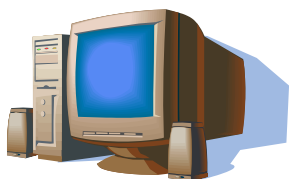
**Decryption Technically OK.  
Business Logic Error.**



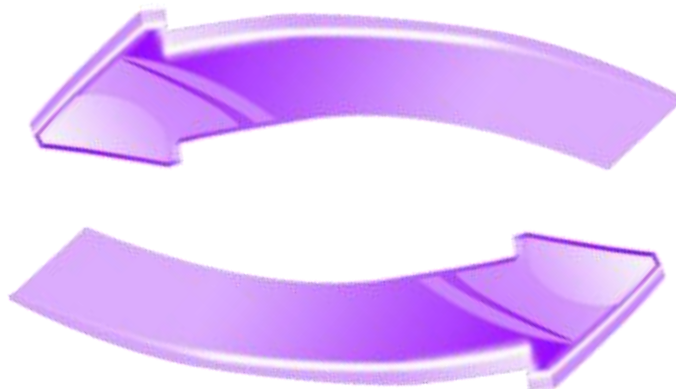
**Error Message Including  
Parameters (Decrypted)**

**Now we know what the  
parameters need to look  
like**

## User Input (Plaintext) – Attack (eg SQL Injection)



Server



User Input (Encrypted)

**Chosen Plaintext Attack**  
ECB mode encryption

**Decryption Technically OK.**  
**Business Logic Success.**  
**Compromise.**



**User Input (Encrypted)**  
**Fed in to another part of the application**

# Vulnerability Chaining

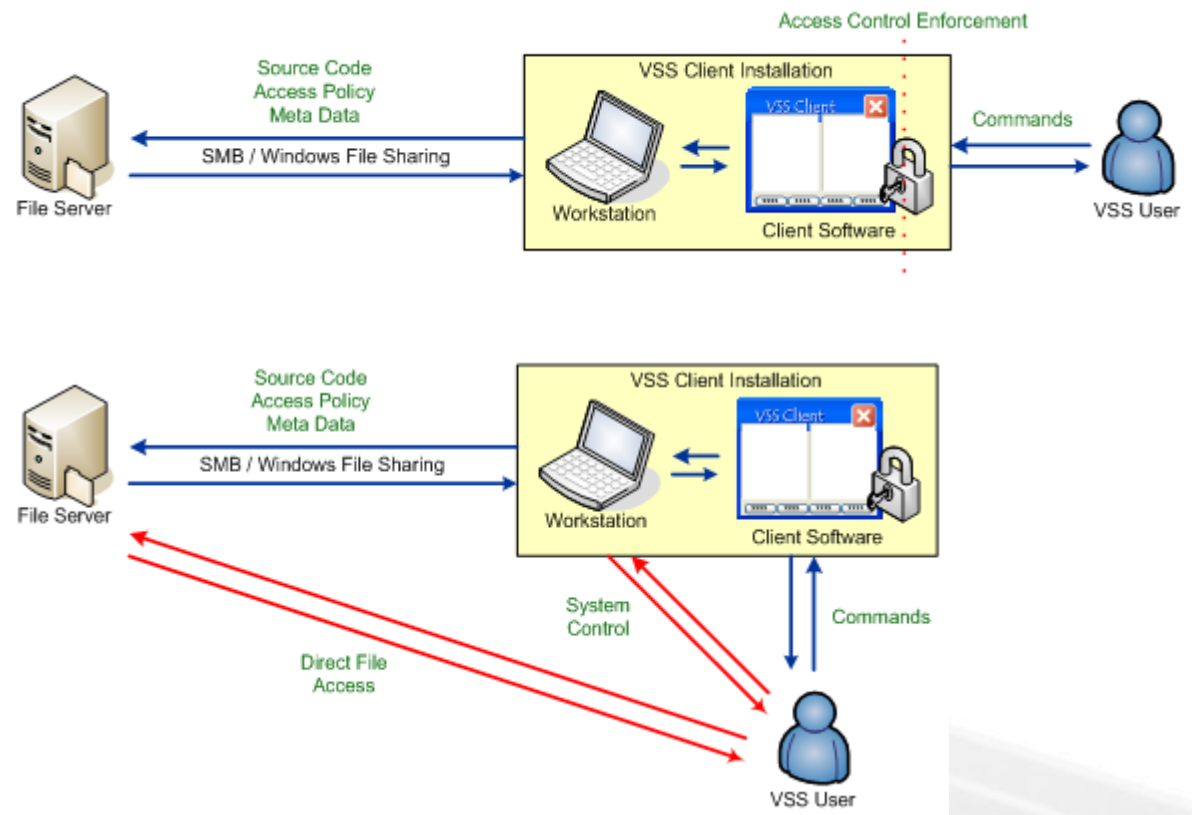
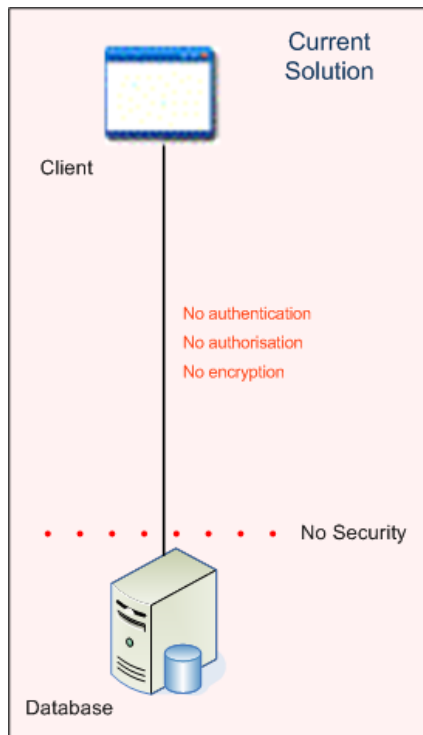
- Using one vulnerability, to exploit another, to gain access with which you exploit another...
- Increasing importance in penetration test style engagements as simple attacks get weeded out.
  - One attack to subvert the control
  - Another attack to obtain privileges
  - A third attack to take control



# Under-rated Attacks

- Cross-site request forgery
  - Endemic.
  - A real threat, but very few organisations are taking the time to resolve it.
- All you have to do is get someone to visit a page with the following html tag in it:
  - ``.
  - How hard can that be?

# Thick Clients



# Thick Clients

- Java applet
  - Client-side authentication... just tell the client it's all ok.
    - result=OK
    - userAccessProfile=YYYYYYYYY
  - Entire database including identity numbers and personal details could be enumerated by Internet-borne attackers.
  - Over 1,000,000 records.

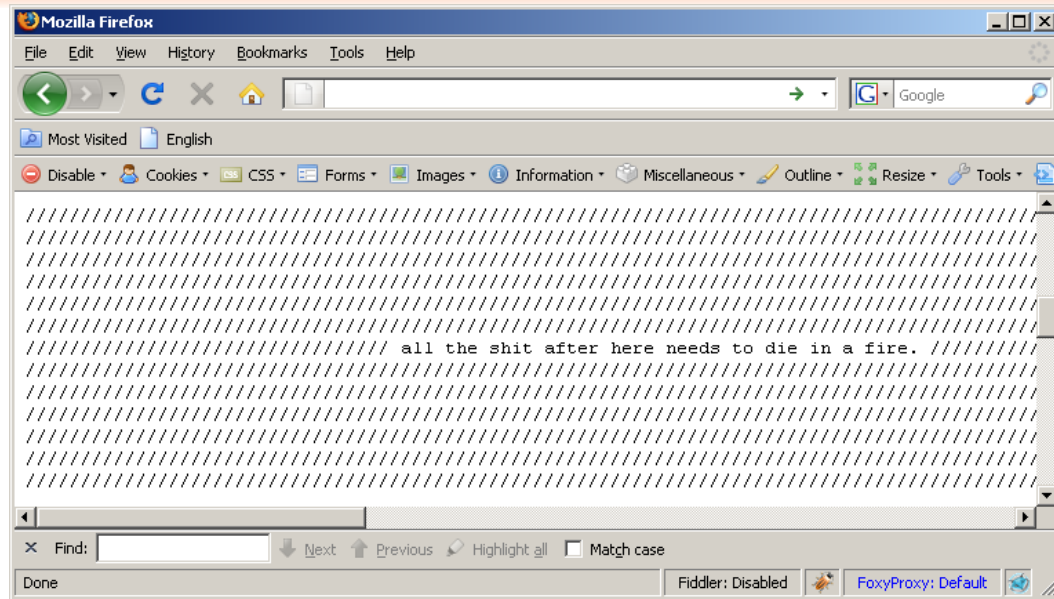
# What have we learned?

- Most people don't understand what we do
- 'Common' != Secure
- Security is not correlated to budget
- Non-Core = Non-Secure
- Most 'solutions' involve papering over the cracks
- Legitimate Access = Game Over

# A lot of people don't understand...

- “Only an authorised user could do that, and we wouldn't authorise a hacker to use the system.”
- “stratsec has made an assumption that the application would be used in a malicious way. Our users are not malicious.”
- “If someone did that, we would sack them.”
- “That wouldn't happen because it requires digital signatures and hackers don't sign their work.”
- “That requires a user to click a link in an email so we aren't concerned about that.”

# ...and a lot just don't care or get rushed



- Found in the source code of an application we were testing...
- I wonder where to look for vulnerabilities
- Other common code comments:
  - to do, fix sqlinjection
  - to do: add encryption here
  - etc

# 'Common' != 'Secure'

- Just because a system is widely used, does not make it secure
  - Core banking systems
  - Internationally recognised ERP software
  - Widely used content management systems
  - Widely used HR information systems
- “Our peers use this, so it must be ok” is dangerous logic

(Don't forget David X. Li's Gaussian Copula Function)

$$\Pr[T_A < 1, T_B < 1] = \Phi_2(\Phi^{-1}(F_A(1)), \Phi^{-1}(F_B(1)), \gamma)$$

# Non-Core = Non-Secure

- There are almost certainly **big** risks, in non-core parts of your operation
  - HR IS
  - FMIS
  - CRM
  - Marketing
- Often outsourced (hence not controlled by your IT team)
- Often very sensitive data

# All Organisations Have Weak Points

- The sheer number of applications maintained by big companies & departments:
  - Most organisations have literally *hundreds* if not *thousands* of owned/managed applications (excluding COTS)
  - Transferred ownership in organisational restructures and changes
  - Unmanaged / unsupported
  - Any concept of budgetary responsibility long gone

# Issue Remediation

- Often band-aid fixes – try to remove the appearance of vulnerability
- You can rarely solve a problem through using the same approach that you used to create the problem
- Re-Test Findings: At least 50% of issues unresolved when we re-test after they have been “fixed”

# Legitimate Access = Game Over

- Rogue Employee Testing for a client in the IT industry
- Primary targets
  - Payroll
  - Director background and financial information
  - Financial results
  - Domain controller
- We were provided
  - An access pass to the office
  - A standard IT domain account
  - A laptop setup for domain access
  - A cubicle

## Schoolgirls fined for bugging teachers

Published: 24 Aug 10 09:17 CET | Double click on a word to get a [translation](#)

 Share 29

 Tweet 18

Two Stockholm schoolgirls have been taken to court for trying to bug their teachers during a grading conference. They were found out after one of them revealed all on Facebook.

- [Red-greens split on pensioner tax cuts](#) (25 Aug 10)
- [A third of Swedish pupils fear school bullies](#) (23 Aug 10)
- [Liberals: Link free school funding to quality](#) (19 Aug 10)

The pair, who are in their mid-teens, came up with the idea after finding a key to the staff common room. They bought basic bugging equipment in a gadget shop, waited until the end of the school day, and planted the device in the staff room.

The girls, who attend a middle school in the capital, planned to listen in on a meeting the following day at which teachers would decide their grades. They were hoping to glean information that would enable them to get their grades improved.

The plan might have gone off without a hitch if one of the girls in her enthusiasm had not revealed all on Facebook, according to Metro. The girls were prosecuted for trespass and arbitrary conduct and fined 2,000 kronor (\$270) each by Stockholm District Court.

TT/The Local ([news@thelocal.se](mailto:news@thelocal.se)/08 656 6518)

# Day 1 & 2

- Organisation chart
- Small number of source code files
- Four (4) domain account passwords
- CFO's personal assistant file server directory
- Employee expense claims
- Corporate credit card numbers and details
- Financial results announcement (prior to ASX release)
- Contents of Active Directory
- Full source code to some products
- Seven (7) domain account passwords
- CEO's personal assistant file server directory
- Service account passwords to a number of databases
- Hardware designs and schematics
- Passwords to various internal and third party systems
- Sensitive documents

# Day 3 & 4

- Account passwords for the production financial & HR databases

```
<connectionString xsi:type="OracleConnectionStringData" name="PeopleSoft Oracle Db">  
  <parameters>  
    <parameter name="password" value="      " isSensitive="true" />  
    <parameter name="server" value="      " isSensitive="false" />  
    <parameter name="user id" value="      " isSensitive="false" />  
  </parameters>  
</connectionString>
```

- Intranet web server control
- Source code for newer products
- Majority of domain account passwords
- Administrative access to security camera system
- Control of content filter server
- Domain administrator access to the corporate domain

- Control of building access control software
- Control of BlackBerry Enterprise Server and management software
- Password to existing domain administrator account
- Passwords to router and switch devices
- Latest & most valuable source code

# Closing thoughts

- Most genuinely critical systems are accessible internally only. (or are *easier* to attack from inner layers)
- It is actually not our system knowledge that would be most useful but our organisational knowledge.
- Once we are ‘inside’, it is easy going.
- Big impact attacks are combining system compromise & control with ‘business level’ fraud.
- Ultimately, this is closely related to the “Advanced Persistent Threat” scenario.