

ColdFusion Security and Risk Management

Justin Mclean

Email: justin@classsoftware.com

Twitter: @justinmclean

Blog: <http://blog.classsoftware.com>



Security

- No system is 100% secure
- Security takes time and effort and can be costly
- How much security is actually needed?
- Is a security feature actually effective?

Risk Assessment

Risk assessment is a tool used to balance business objectives and security requirements in order to achieve cost effective security measures.

Process

1. Identify assets
2. Identify and quantify the possible threats
3. Determine the consequence of each threat
4. Evaluate the current risk
5. Decide acceptable level of risk
6. Treat each risk

Frequency

- This is not a once off process
- Repeat when:
 - System is in place
 - Changes are made to the system or assets
 - Made aware of new threats

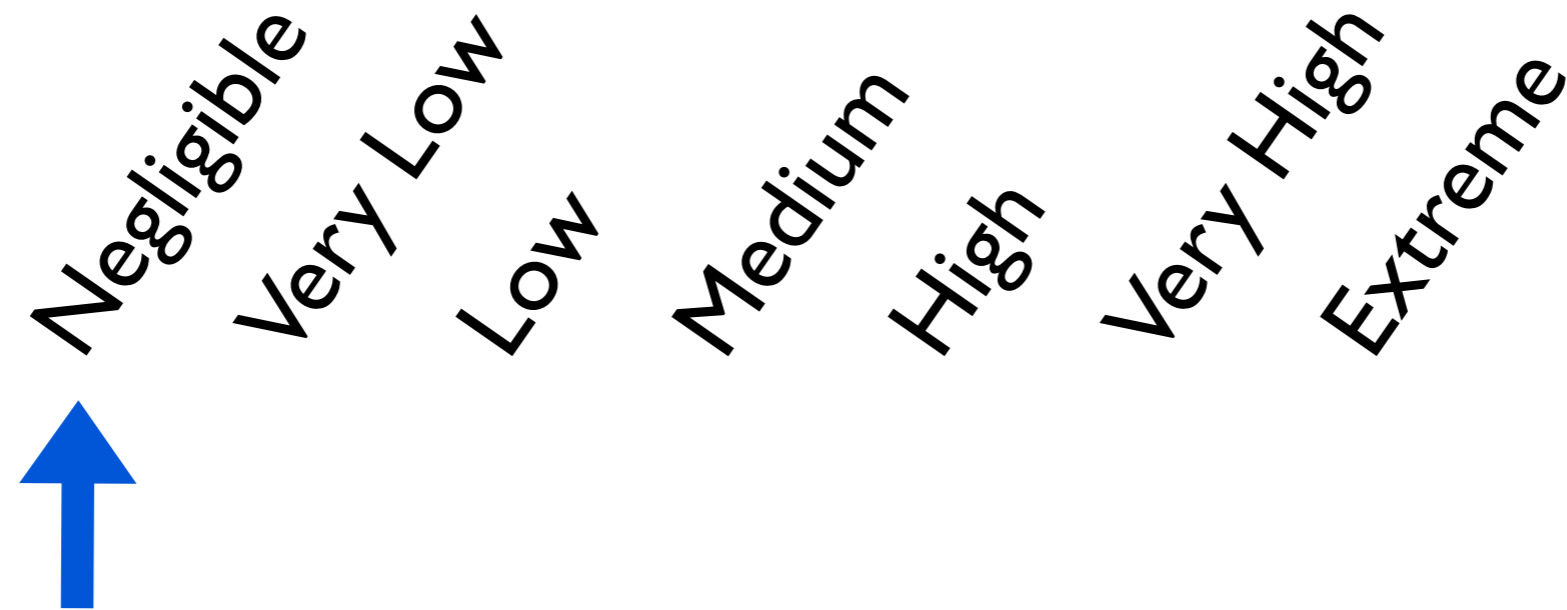
Identify Assets

- Create list of assets
 - Hardware
 - Availability of service
 - Integrity of information
 - Reputation of system and/or organisation
 - Staff

Identify Threats

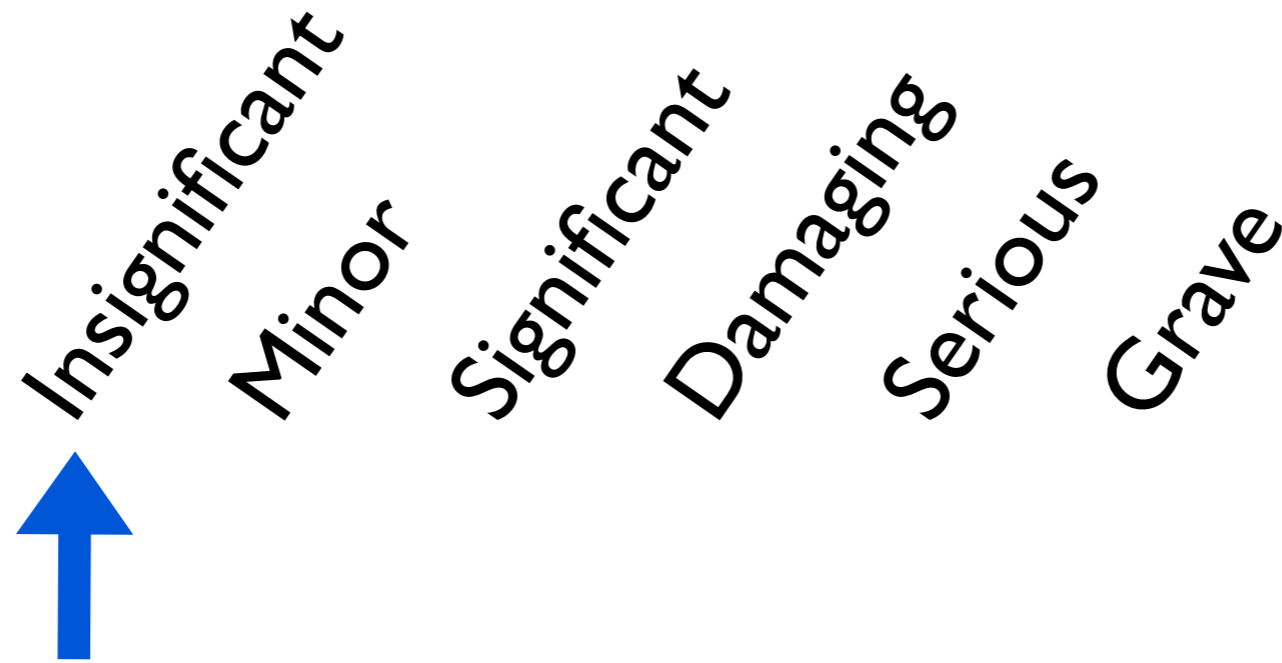
- Create a list of threats
- Be creative include unlikely threats
- Tendency to ignore obvious threats
- Careful of preconceived attitudes

How likely is each threat?



Unlikely to occur

Consequence of each threat?



No extra effort to repair

Risk

- Risk is a combination of frequency and consequence
- The more likely a threat will occur increases risk
- The more often a threat will occur increases risk

Risk

Consequence

Threat

	Insignificant	Minor	Significant	Damaging	Serious	Grave
Negligible	Nil	Nil	Nil	Nil	Nil	Nil
Very Low	Nil	Low	Low	Low	Medium	Medium
Low	Nil	Low	Medium	Medium	High	High
Medium	Nil	Low	Medium	High	High	Critical
High	Nil	Medium	High	High	Critical	Extreme
Very High	Nil	Medium	High	Critical	Extreme	Extreme
Extreme	Nil	Medium	High	Critical	Extreme	Extreme

Acceptable Risk

- Set level of acceptable risk
- Assign priorities for each threat based on acceptable risk and risk of threat

Priorities

- A required risk $>$ actual risk + 1 levels
- B required risk = actual +1 level
- C required risk = actual risk
- D required risk $<$ actual risk

Treatments

- Minimisation of harm
- Transference of risk
- Acceptance of risk
- Addition of security measures
- Change of service or system specifications
- Reduction of security measures

Effort and Cost

- May be many ways to treat a single threat
- Amount of effort or cost may decide which treatment chosen

Election System

Student election system for the
University of Technology Sydney



UNIVERSITY OF TECHNOLOGY SYDNEY

UTS: ELECTIONS ONLINE POLLING BOOTH

THINK.
CHANGE.
DO

UTS Online Polling Booth

You are entering a secure system with time restrictions. If you are timed out you will need to log in again.

Your vote is confidential. Login details used for system access are separated from your voting choices.

You are bound by the terms established by UTS Policies, including the [Information Technology Security Policy](#) and [Acceptable Use of Information Technology Facilities](#). UTS reserves the right to take disciplinary action in any case of misuse, or attempted misuse, of the Online Polling Booth or other breach of UTS policies.

Authorised voter log in

User ID

Password

Students use your UTS student webmail account user ID (8 digit student number) and password.

Staff use your UTS webmail/LDAP/neo login (username or 6 digit staff number) and password.

For log in help, contact IT Service Desk on (02) 9514 2222 or submit a service request at <https://servicedesk.uts.edu.au>.

Contact
Elections inquiries
e: elections@uts.edu.au
University of Technology, Sydney
15 Broadway
Ultimo NSW 2007
Electoral Officer
University of Technology, Sydney
P.O. Box 123
Broadway, NSW 2007
Australia

Student Election System

Server Configuration

- Run with minimal down time
- Perform well under load
- Limited external access to server

Server Treatment

- Staging/production system
- Standalone separate machines for database and CF server
- No access to production server
- Code reviewed by external agency

Network Issues

- Occasional network outages
- Occasional slow access from outside

Network Treatment

- Ability to change the end date after an election has started
- Date could only be moved forward ie extended not reduced

SQL Security

- SQL injection attacks
- Sensitivity of data
- Trust and integrity of election results

SQL Security

- Multiple data sources
- Multiple database users
- Restrict SQL actions. No deletes and almost no updates few inserts and mainly selects.
- Table level permissions

Disable Connections	<input type="checkbox"/>	-- Suspend all client connections.
Login Timeout (sec)	<input type="text" value="30"/>	
CLOB	<input type="checkbox"/>	-- Enable long text retrieval (CLOB).
BLOB	<input type="checkbox"/>	-- Enable binary large object retrieval (BLOB).
Long Text Buffer (chr)	<input type="text" value="64000"/>	
Blob Buffer(bytes)	<input type="text" value="64000"/>	
Allowed SQL	<input checked="" type="checkbox"/> SELECT	<input type="checkbox"/> Create <input type="checkbox"/> GRANT
	<input checked="" type="checkbox"/> INSERT	<input type="checkbox"/> DROP <input type="checkbox"/> REVOKE
	<input checked="" type="checkbox"/> UPDATE	<input type="checkbox"/> ALTER <input type="checkbox"/> Stored Procedures
	<input type="checkbox"/> DELETE	
Validation Query	<input type="text"/>	

Datasource Options

Multiple Database Users

- Table level permissions
- SQL operation permissions

SQL Permissions

- Deny all to all users to all tables
- Add permissions for each SQL operation as needed
- Don't be tempted to give admin user all permissions

Deny All

deny **all on** elections to electionvoter,
electionadmin, electionlogin

deny **all on** candidates to electionvoter,
electionadmin, electionlogin;

deny **all on** rolls to electionvoter, electionadmin,
electionlogin;

deny **all on** ballots to electionvoter,
electionadmin, electionlogin;

Grant Access

grant **select** on roll to electionvoter,
electionadmin;

grant **update** on roll to electionvoter;

grant **insert** on roll to electionadmin;

Login Issues

- Dictionary attacks
- Timing attacks
- Storing passwords

Login Treatment

- Account lock out if password wrong x times
- Random time delay

Java Sleep

```
<!--- delay is to hinder timing style attacks --->  
<cfset thread=createObject("java","java.lang.Thread")>  
<cfset thread.sleep(300 + int(rand()*21)*10)>
```

Code Modification

- Check for modified ColdFusion code
- Only run trusted pages

Code Treatment

- Finger print each page via MD5
- Check finger print when page is run via Application onRequest method

onRequest

```
<!-- read the cfm file -->
<cftry>
  <cffile action="read" variable="pagecontents"
file="#CGI.PATH_TRANSLATED#">
  .....
</cftry>

<!-- get page from database -->
<cfquery name="dbpage" datasource="#request.datasource#">
  select page, hash from pages
  where page = <cfqueryparam value="#hash(listlast(arguments.page, '/'))"
#" cfsqltype="cf_sql_varchar">
</cfquery>

<!-- check if page exists and page hash is correct -->
<cfif dbpage.recordcount is 1 and hash(pagecontents) is dbpage.hash>
  <cfinclude template="#arguments.page#">
<cfelse>
  <cfinclude template="./elections/security.cfm">
</cfif>
```

Limit Information

- Assume someone will break into the system
- What information can they obtain?
- Minimise damage they can do
- Log everything

Why do this?

- Know that you've spent your budget where it's needed when it comes to security of your system
- Confidence in your system is secure as you need it to be
- An understanding of the risks in your system
- Minimal damage and a plan of action if the worse does happen

Questions?

Ask now, see me after the session or email me at justin@classsoftware.com