Interested in learning more about security?

SANS Institute
InfoSec Reading Room

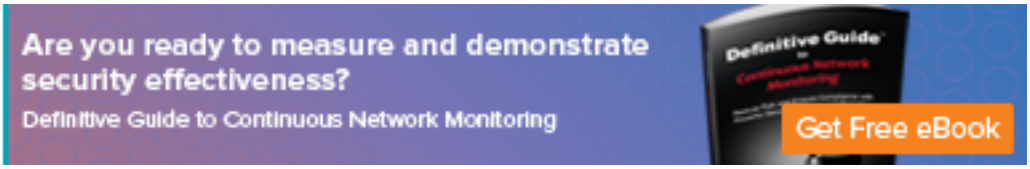
This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

A Paper on the Promotion of Application Security Awareness

Application security is not a new science and the same principals that apply to network security also apply to application security. The cause of most data problems is due to a lack of management concern. The security focus should be a managerial issue rather than a technical issue. Hence management should clearly establish and enforce security policy, plans and procedures, to guard against costly and embarrassing breaches of security.

Copyright SANS Institute
Author Retains Full Rights

AD



A Paper on the Promotion of Application Security Awareness

By

Man-Sau Yi

August 2001

GSEC
Version1.2e
ID: Man-Sau001

© SANS Institute 2001, Author retains full rights

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

1.0 Introduction

"With intellectual capital the creator of wealth in this century, it is imperative for companies to protect themselves from threats of mis-use, abuse or theft of their sensitive information"[1].

When we think about the security stories that make the headlines they are often about the compromised web sites, the lost assets and the hackers bribing a multinational company.

What does not make the headline news is the compromised application. The very application that holds the information, the jewel assets of the company that could have stolen the headlines. Some how the application security does not make interesting reading material.

Application security is not a new science and the same principals that apply to network security also apply to application security. The principals are:

- Know your application
- Defense in depth
- Principal of least privilege
- Prevention is ideal, though detection is a must.

The cause of most data problems is due to a lack of management concern. The security focus should be a managerial issue rather than a technical issue. Hence management should clearly establish and enforce security policy, plans and procedures, to guard against costly and embarrassing breaches of security.

The emphasis on management involvement is not totally unrealistic as a security minded company comes at a cost to the bottom line. And if a publicly quoted company is under constant scrutiny by the analyst and by the shareholders about its 'performance in the market'; cost becomes a serious consideration and with management controlling the cost line this would have a direct impact on the size of the resource. A balance will need to be struck between cost and how much security focus the company needs to maintain.

2.0 Security – begin at the beginning

The analogy that is often referred to is:

"If security were the electrical wires then it is more difficult to put the wiring in after the house is built."

The old school of thought is "build it first and fix it later'. Security issues are often the last issue that gets addressed in either development or testing, and fixing a security issue is not even worth a mention when budgets are tight and a project over runs.

A cure to this malady is a douse of Zen wisdom that is encapsulated in one of the Zen principals, which states, "A change in one's own mind pattern is enough to induce the necessary outcome without changing your environment". In this context it

[1] ISP Innovative Security: Microcomputer Security: www.isecure.com/pc-security-white-paper.htm

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

would mean: "Why not incorporate security at the start of a design plan even if you cannot stop either the internal or external hackers?"

3.0 What is your plan?

Application development is a fine balance between functional requirements and business needs; a fine balance between resource and deadlines; and a fine balance between security and risk. The trick is to figure out whether the balance has been met and could we live with the remaining imbalance(s).

A useful tool to assist in determining and measuring the degree of balance is the development plan. Between the choice of a spiral plan and the straight-line plan, the spiral is preferred. Why? With a spiral plan there is scope to review and refine a security issue against a functionality requirement, for example. The plan allows for the process to be repeated and ensures that the security functionality is refined as the application is built; thus avoiding a one off gun shot approach, which is a feature of the straight-line plan.

Below is an example of a development plan with security in mind and it is less driven by the functionality's deliverables:

- Identifying the security requirement and the risks
- Design the architectural design
- Secure implementation
- Attack stimulations, and
- Maintaining the application.

4.0 Knowing your application

Do you know your application? A good starting point is the technical specification. Though in the real world how many newly built application and legacy systems are supported by either a technical specification or the original specification with all the change requests, respectively? I leave you to mutter the answer to yourself.

Understanding the coding that was used to build the application meets one of the four security principals "to know your application". You do need to get intimate with the coding. Why? Each programming code has unique features. The general approach has been to code an application in the latest language with little understanding about the code's capabilities and limitations. With each code there is a trade off between efficiency and security. C and C++ provide efficiency and speed, though the code pays little attention to security issues. With Java, the coding language has more security aspects but its limitation is that the deliverable is not fast relative to a C and C++ program.

Knowing what could go wrong with coding is important in that they affect security issues. Here are some of them:

- the dreaded buffer overflow (the most popular security problem today)
- format string could also cause over flow problem for hacking
- race conditions re distributed computing.
- Random number generator e.g. RAN are not random and can break down crypto.
- Crypto misuses.
- Trust problems i.e. too trusting on the input coming though.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

- Authentication problems

As part of getting to know your application you could read through the code. However, reading code is not only time consuming but it is also prone to human errors. All is not lost as there are user's automated tools out there to scan C and C++ source code for buffer overflow and race conditions. There is no need to go over board in using such a tool. Strike a middle ground between accuracy and efficiency. Know what you are trying to deliver.

An example of a code-scanning tool is ITS4. ITS4 technical features include

- break file into lexical tokens
- match patterns against known vulnerabilities (stored as rules)
- why not a real parser? Avoids false negatives and makes the tool run faster.

With all such tools ITS4 does have its limitations: static analysis of C is harder; C++ is even harder

None-the-less there are good reasons to use ITS4: -

- Programmers may not know much about software security
- Programmers may not understand the dangers
- Programmers are not sure how to fix potential flaws
- Programmers take the easy road

5.0 Technical specification is a good idea?

Time often comes at a premium when there is an application problem to remedy. At the very least have a technical specification written up and have it continuously updated as the application goes through the development cycle. Why? It all seems like a lot of work with no benefits. However, there are a lot of reasons to support a technical specification. So what are they?

1) How could you test an application if you do not know what you have built?

   Unless the application is not complicated there is often more than one programmer on the development site. Beside that, there would be many alterations to the prototype before the finish product is in production.

   Often than not, test plans are developed towards the end of the development cycle. What may happen is that the actual development takes longer than expected and the testing time is squeezed with little time remaining to write and implement a proper test plan. With the time squeeze, the test plan may not reflect the final product. To resolve this gap issue an up to date work-in-progress technical specification would be the first reference point in framing a test plan, which identifies the key test areas. By having the specification available this makes the test time both effective and efficient. An ideal? It actually works.

2) No one design things for the wrong reasons. It would be worthwhile to get an independent review by some one other than the people in the coding team. The organization may have a security risk team and this is where they could provide additional resource to the project. S/he doesn't have to be an outside expensive consultant. Why not ask them to contribute to the security aspects

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

of the system during the build cycle. Regard the security personnel as a resource and not a burden to the process. Remember two heads are better than one. The technical specification is a vital piece of documentation for this type of analysis.

6.0 Analyze the software requirements

So far, we know that before we even start coding we need to appreciate and determine the balance between efficiency and security in the coding program; and to have a development plan that incorporates a review cycle and a work-in-progress technical specification to make sure the security controls are considered and tested.

So what are some of the application security issues? Ten principals are listed below. Again, strike a middle ground of what you are trying to deliver and keep it simple (see principal eight, below).

Principal one - Identify and secure the weakest link.

- √ Identify the risk and rank them.
- √ Perform a cost/risk analysis of securing the risk.
- √ Though you need to identify what to protect, from whom and for how long.

Principal two - Practice defense in depth

- √ Make sure there are multi-levels of access security from application access to access to sensitive information;
- √ Lock out of the application after a period of inactivity;
- √ Provide additional security over sensitive information; and
- √ Apply the Principal of least privilege and ensure the users only have access and functionalities to complete their job.

Principal three - Be reluctant to trust

- √ Identify the trust relationship between each component;
- √ Identify the risks arising from the data flow;
- √ Practice the Principal of least privilege; and
- √ Get an independent resource from security risk management to perform a high level review on the risk (not the functionality).

Principal four - Remember hiding secrets are hard

- √ Security through obscurity doesn't work.

Principal five - Follow the principal of least privilege

- √ Give the users only the access that they need to perform their role. There may be a need to build a module segregates the duties and privileges.

Principal six - Fail and recover securely

- √ Make sure the application is linked or has its own detection and appropriate level of alarms system to alert invalid access (internal or external);
- √ Check the logs;
- √ Keep the audit trail, as this is vital evidence if the issue has to be taken to court; and
- √ Have a proper back up copy of the application and the data, as a full recovery may be required.

Principal seven - Compartmentalize

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

- √ Limit invalid access (internal and external) to assets;
- √ Make sure the IT support team are not the same team who backs up the system; and
- √ A segregation of duty review is a must.

Principal eight - Keep it simple
- √ Avoid ambiguity and hidden assumptions; and
- √ Keep the coding consistent

Principal nine - Keep trust to yourself (reverse of social engineering)

Principal ten - Be skeptical
- √ Questions all assumptions and choices in the code, for example; and
- √ Make sure the code could be testable.

7.0 It's a lonely road home

We now have a developer who ensures security controls are considered up-front before the application hits the home plate. Problem? Possibly, s/he is probably one of a few developers who share this approach. The road a head may appear bleak if you are the only one following this path.

Often management becomes aware of the importance of security issues is when the company's systems have been compromised. The attack should not necessarily be considered as bad publicity as long as management remedies the situation rather than assume that the company's assets would not be compromised a second time.

There are many ways that management could provide support and one of them is to put in place a security/information risk department. The primary function of the department is to ensure that securities levels are raised and are met. The level of commitment is set out in the policies and standards that the department is also responsible for creating. A further support by management is to approve the policies with in a 'risk committee', whose primary function is to provide the status quo for ensuring that the company meets the desired security level in all the business' undertakings.

The policy and standards should act as enforceable guidelines to establish a similar playing field for all applications. Though bear in mind that when setting out the policy and the standards the level of compliance should justify the deliverable.

For an application policy the following points should be considered:

- Development, acquisition and change phase;
- User identification;
- Password setting and authentication;
- Access control (privilege and restriction);
- Application accountability (controls in place to ensure accuracy of data and guide against unauthorized modifications);
- Log on/log out;
- Security over operational support (IT support and other IT functions, like back ups should be segregated);
- Monitoring and incidents (audit trails available by which to determine whether security breach has occurred.);
- Data exchange (securing transmissions over communication channels); and

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

- Back ups and contingency plans.

Other pointers to consider when designing application security policy:

- Who should be involved;
- The resources that we are trying to protect (identify your assets);
- Which people do you wish to protect the resource from (note that most security breaches happens internally);
- Risk assessment of the potential risks;
- How important is each resource;
- How regular should the application re-assessment be to endure that the objectives and the application security have not changed;
- Does the policy comply with the law?; and
- Most importantly, is the policy practical, workable and enforceable?

8.0 Awareness

What's next? The Risk Committee approves the policies and standards and they are posted to the intranet. The IT department is backed up by management's unified approach on the importance of application security. Hey guess what? Have you browsed through the security risk management's web site lately?

Awareness is key to improving security awareness for everybody in the company. The audience covers the whole company; no one is immune from receiving regular updates and security awareness training.

Make sure the information is disseminated at the level that befits your audience's background. Also the method of disseminating the information is important. For instance, a presentation would suit some groups whilst others may prefer to access the information via the intranet.

Some companies have a new starter's orientation course. This is a great opportunity for a member of the security risk team to bring the security awareness to a new employee on their first day. Going forward, the employee should be encouraged to attend any internal security awareness training.

A stronger recommendation is to ensure that all users sign a declaration that states they have read, understood and continue to comply with the security risk management policy for the company. How often? Annually and it should also be signed on day one when they join so they are aware right up front of their duties.

9.0 Handling security violations and breaches

How security breaches and violations are handled depends upon the business. The business should define the actions to be taken based upon the different types of violation, starting with whether the breach was performed internally or externally.

The action could range from a written warning to filing formal legal charges. Whatever the approach is, an approach should be in place so that prompt and decisive actions are taken if a security breach occurs.

Again, make sure this information is widely disseminated and publicly available to avoid all ambiguity.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

10.0 It's the cycle of life

Ever heard of a company's standing still? If you have its likely it will not survive much longer. Today's business is faced with constant changes and meeting the demands of their consumers, their shareholders, the market analysts, the predator and if not get ahead of their competitors, at least stay with the market. This momentum of change requires continuous evolution of their products, their image and not least of all the technology.

With all these makeovers and changes it is just as important that the security is not compromised with upgrades, patches and replacements. Security risk department should provide support though continuous assessments and up grading the policies and standards that reflect the latest security issues. Training and awareness should be effectively communicated to all staff members.

IT development should ensure that security is not compromised with up grades and patches. Change management should play an important part to ensure that the security is tested before the application is released to the production environment. The test plans should be as rigorous as the development test plan. The message is "Security should not be compromised".

Everybody should have a part and they should play their part to ensure that the company not only stays competitive but also everybody should keep an eye on the big bad wolf.

© SANS Institute 2001, Author retains full rights

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Information Resource

- Exodus Communication: Application Security Code review
  www.exodus.net/security/application_security/code_review.html

- Citgital Inc (In house presentation on web based applications)

- The white paper on Back to Information Security basics
  www.bestweb.net/~bgeiger/art_paper/wp_backbasics.html
  There is a problem with this web site. I have attached a copy of the document here.

  yi.doc

- ISP Innovative Security: Microcomputer Security: www.isecure.com/pc-security-white-paper.htm

- Application program security: handbook of Information security Management
  http://www.secinf.net/info/misc/handbook/003-006.html#Heading1

- Purposes of Information Management Security:
  http://www.secinf.net/info/misc/handbook/019-021.html#Heading2

- Application development:
  www.atstake.com/services/enterprise/applications.html

© SANS Institute 2001 - Author retains full rights

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 5th, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
Security Leadership Summit & Training	Dallas, TXUS	Dec 03, 2015 - Dec 10, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced