

SUCCESSFULLY MANAGING A WEB SINGLE SIGN ON PROJECT

This paper is designed for senior managers wanting to know what their expectations should be of their WSSO project manager, consulting staff and project team during a WSSO implementation. It briefly touches on some of the key points to watch for when managing a web single sign on (WSSO) project in order to minimize risk, costs and time to deployment while ensuring a sustainable infrastructure by the enterprise staff after the consultants and product vendor specialists leave.

Copyright, 2002. [Guy Huntington, HVL](#).

INTRODUCTION

Deploying a Web Single Sign On (WSSO) project in an n-tier environment is akin to conducting a game of chess on multiple levels. A change in one tier can affect all the others. What looks simple therefore on an executive PowerPoint presentation extolling the virtues of WSSO, is very complicated in real life to implement and sustain.

This paper is aimed at those who are either contemplating such an initiative or, those folks left holding the bag to deploy after the executive decision to proceed has been made. If your WSSO initiative involves only one application, a single operating system and the same type of web or portal servers, you may not need this paper. BUT, if you're in a large enterprise environment with multiple NOS's, web servers, portals, reverse proxies, different hardware platforms, multiple identity data stores and multiple applications to integrate into WSSO, read on.

REALITY CHECK

Contemplate for a moment your large WSSO deployment a year or two from now down the road when a majority of your applications are using it for authentication and various forms of authorization.

What would happen if your WSSO didn't work for a minute, an hour or several hours? Your employees, customers and business partners would not be able to access or use any application dependant upon WSSO. What impact would that have on the enterprise's productivity, the enterprise bottom line, customer and business relationships etc.?

It's with this sobering thought that I approach a WSSO project. WSSO is an incredible tool if well designed and implemented. Poorly executed, it has the potential to create single points of enterprise application failure and/or security breaches with potentially disastrous implications.

If you want your project to succeed, there are several areas within your WSSO project you need to carefully review and manage. These include the identity, authentication, authorization, session management, auditing and infrastructure maintenance.

IDENTITY

The identity is the bedrock of your access system. You might be surprised to find out there's work to be done in cleaning up account management to provide identity integrity.

If you have multiple identity stores within your enterprise (network accounts, ERP, reverse proxy, CRM, different LDAP stores, etc.), get down to the nitty-gritty and ensure that the unique global ID you're providing to an identity is really unique. You need to be 100% confident that there is a one-to-one mapping between a unique ID and a real person.

There is a significant and clear-cut difference between what is a real person and the roles that a person may play. You don't want the same person to have a unique ID for every role that they play.

Within many large enterprises there will be confusion over the definition of what constitutes an identity, role, accounts and the unique ID used for these. In some enterprises, an individual may assume several identities with each having one or more roles along with unique ID's for the roles and/or the identities.

Your job is to first get people to agree on a common set of definitions for accounts, identity, role, global unique ID and network ID's. This will quickly help you figure out what people mean when they use these words with respect to the systems they administer and use. Without this, you can waste a lot of time and effort due to many people making false assumptions about what other systems do.

If you're using network accounts as an identity repository for authentication, determine how test accounts are used. Look for application areas such as help desks where people may need to assume an identity on behalf of someone else. Watch out for people possibly sharing accounts, thus assuming the same role or identity.

A WSSO project crosses many different identity stores usually under different departmental and system administration. Many of these systems may not communicate well or at all with each other in providing identity updates (they may never have had to in the past). As a result, you may end up creating a WSSO system that still allows an identity into the applications months after that person whom the identity maps to was terminated, promoted, etc.

In order to avoid this, you need access to the data structures for the identity repositories you're going to be using as well as maps of the business processes that create and update the repository data. This may be hard to obtain and/or take time for identity administrators to give it to you. Be wary of high-level assurances from the identity administrators that everything will work. The devil is in the identity details!

Determine in advance if you are going to need to accept and trust an identity being created and presented from outside your own systems (partner's portal, SAML identity, etc). The same applies for identities you'll be creating that your business partners or others may have to trust. Managing the identities may not be straightforward for your enterprise, your business partners or even your customers. Do test pilots well in advance to prove out that identity exchange between systems not only works but also has identity integrity as time passes.

Successfully Managing a WSSO Project

It is often the case that a WSSO project leads to or, stems from, an identity project. Synchronizing and managing the identity stores is not trivial. Managing the identities across the enterprise and outside the enterprise requires a solid set of identity management tools.

Do your tools have the ability to delegate identity administration to the most cost-effective management level(s) without a lot of coding? Equally important, make sure that you can lock down the security separately on each attribute within the LDAP directories. You need fine grained security control for each piece of the identity to maximize manageability while reducing risk.

If you don't do a good job of sorting out the identities, your project may hit the reefs of authentication and authorization. Having a thorough understanding of identities, roles and real people is critical in creating a successful WSSO system. Without this you may be opening yourself up to administrative nightmares and/or potential security breaches. Equally as bad, you may find that you have major problems requiring significant re-working of your WSSO system. This may involve large amounts of additional, unplanned and unbudgeted time, money and resource allocations.

AUTHENTICATION

Authentication requires careful planning and thought. The starting point is some form of risk assessment by the enterprise to determine what strengths of authentication will be acceptable for different applications. For instance, a username and password may be acceptable risk for authenticating an identity for some applications but considered insufficient strength for others where a digital cert, biometric, tokens or combination thereof may be required.

You need management agreement on what constitutes acceptable risk or, you may find that a manager of a particular application or entire business unit may refuse to work with WSSO since they believe they cannot trust the authentication being presented to them. This can significantly affect and/or alter your project timelines while also creating unnecessary political problems for you to manage.

Next, depending on the WSSO tools you're using, you must decide on a structure for differentiating the levels of authentication as well as implementing them. The tool needs to be configured for different authentication levels as well as allow room for future technologies and changes to enterprise security policies.

Then there's the major issue of how to deploy the authentication methods with the WSSO, network and web server infrastructure. For example, you may decide you never want passwords to travel in the clear. Thus, you may want to divert a request for a web resource from a web or application server to a WSSO authentication server that uses SSL, TLS etc. After successful authentication, you'll want the option to either continue on with a secure connection to the user's browser or, redirect back to an unencrypted web or app server. There are cost and performance implications from doing so that you need to carefully think out and plan for.

Have you thought about what happens with unsuccessful authentications? How much information is passed to the help desk or user, if any, to let them know why the authentication failed? Do you have plans to integrate password management with the WSSO system? Are you using online verification of digital certs? Will the digital certs work with different browser versions? What are the use cases and test parameters for determining there are no PKI/cert issues? What are the token management/WSSO issues? What are the biometric standards you're going to adhere to? Is there an API in the vendor's toolkit to deal with this or, are you going to have to do customized coding?

Have you considered how you'll set up multi-factor authentication using your WSSO tools? Can you do weighted authentication where you trust one type of a multi-factor authentication more than another? Can you create a trust formula to calculate a final value required for successful authentication?

Many WSSO systems use non-persistent, encrypted cookies. These are placed on the user's browser to maintain state as well as let the SSO system know what level of authentication has been successfully achieved during the user session. You need to determine if your enterprise browser policy will accept this. Other WSSO systems will use application or SSL servers to help maintain state. Have you examined the cost, performance, installation and maintenance issues of doing so?

In some large enterprises there may be several authentication mechanisms run under the auspices of different departments, business units or system owners who are deeply entrenched. Instead of using only one authentication method for sign on, you may be forced to design a web based SSO solution that gives users the choice as to what authentication method they will use. In other words, the politics may force you to absorb the existing authentication methods so application owners don't have to change much. This can be very time consuming to do, complicated, costly and expensive to maintain.

If you're heading down this path, make sure you have adequate budgets and the right human resources and software tools to accomplish this. There may be a fair bit of politics involved as you gain the different business unit and application owner support.

Regardless of what type of single sign on project you do, an area either often overlooked or underrated by project sponsors and project managers is the amount of education needed to make WSSO successful when it comes to authentication and authorization. You may find your project under intense political pressure, cost increases and time delays from several areas.

With poor WSSO user community education, you may find that some new WSSO users will react poorly to change. As a result they may carry their unhappiness right up to the senior executive suite. You need excellent end user and executive team WSSO education and communication. If you do a good job of this, the executives can help dispel complaints rushing your way which otherwise might derail or slow down your project while consuming unplanned for project resources.

At the same time you're planning the end user education, you also need to get on board the application owners into WSSO. Many application owners in your enterprise will be leery of a new authentication and authorization system. It's usually complicated to explain because of the many n-tier components used such as the security agents, how they redirect to your security servers, when and how they redirect back, etc. Therefore, you need a well-planned major education campaign with application owners on how your SSO authentication works in order to get their buy-in.

A potential minefield to be aware of concerns authenticating real people having multiple identities and roles. As was mentioned earlier in the identity section of this paper, you need to ensure you're authenticating an individual and not a group of people having the same role, identity or network account. There has to be a one to one mapping between the unique ID and the real person you're authenticating.

This problem becomes exasperated once you start sharing and trusting identities, authentications and/or authorizations between systems and enterprises. System administrators will want to know and trust that when you pass them an identity, it's the real person they think it is and not potentially more than one person using the same account.

There's a lot more to consider than what's been outlined above. For example, how are you going to handle wireless authentications? Are you going to encrypt this traffic? What's the performance and cost impact from doing so? What happens with reverse proxy authentications? Does your WSSO system need to differentiate for an application if the user is internal or external to your network?

Have you thought about application-to-application (app-to-app) authentication? If you're like many enterprises, you're either starting to do this already or, rapidly moving in this direction. There are many things to consider which PowerPoint slides extolling the virtues of SSL, TLS, SOAP, XML and web based security services may leave out.

You need to determine how the applications are inventoried and managed within the enterprise LDAP directory. Are you giving each application a unique global ID? In a large corporation with several hundred or thousand applications, this alone is a challenging task to set up and maintain.

Next you need to determine how you're going to secure the connection between the app-to-app. It may be you're going to use SSL or TLS. This will involve the use of PKI digital certs on at least one end of the handshake. You're going to need to determine how the certs are managed, updated and what constitutes the responsibilities of the application owners in this respect, etc.

Then you need to determine how the central WSSO infrastructure is going to work with the app-to-app authentication and authorization. Are you creating a web app-to-app authentication service? Do you require customized code for the software to make all the magic work or not? You'll need to do extensive testing on the service before you announce to the enterprise you're ready for business.

These are just a few of the many, many questions and details to be followed up in creating a successful WSSO authentication system. Stitching the authentication system(s) together with the identity system and integrating to the authorization and auditing systems in a large enterprise is often not trivial. It may take a lot longer and consume more resources than you originally planned for.

If you're in an enterprise that is tightly managed from a core enterprise IT group, you might be able to ignore some of the issues raised in this section. If however you're in a complicated enterprise with many groups of stakeholders running different systems, be wary of glossy assurances from vendors, consultants and your staff re how easy the authentication integration into WSSO will be to accomplish until you've done your homework.

AUTHORIZATION

Authorization in WSSO begins with some form of post-authentication action. Once a successful authentication has taken place, the WSSO system needs to make a decision on how to proceed with authorization.

You need to map out the use case scenarios for authorization. This can be very complicated and/or time consuming. You and your WSSO team need to have very detailed discussions with application owners to understand exactly what is required to mesh their system to yours. In cases of heavily customized or homegrown applications, this can be very complicated, time and resource consuming.

The use case scenarios should include a firm understanding of the management models required. For example, delegated administration of the authorization may be required to extend to several levels of the enterprise. Are you going to use groups, roles, titles, position, geography or a matrix of these variables to determine an authorization? Is this information in your LDAP directories? How easy is it to work with? Who maintains it? How is it updated?

Equally important is how a change in authorization models and/or applications made in one part of the enterprise makes its way into the WSSO system. If it's cumbersome or time-consuming for an authorization change to make its way into WSSO, you may be setting yourself up for potential security breaches. People will find ways to work around cumbersome processes.

You also want to focus on authorization exceptions. The reality is application security management is full of exceptions. Someone will need access to a system for which they normally would not be granted access. When an exception needs to be made, who decides? How does it affect WSSO? How is the change made? Who makes it? How easy is it to make? How is the change tracked? What's the quality assurance process for this?

Authorization in large enterprises having many applications is usually very complicated. Beware of simple sounding solutions or the belief that the WSSO "product" will somehow solve all your authorization challenges with little modification. Oftentimes, changes need to be made to business processes, management models and code in the WSSO product and/or the applications.

You'll need to focus in on testing and management of the WSSO authorization rules. Long before the authorization rule is implemented in the production system, you will want to know that extensive testing and debugging is done on the rules, otherwise potential security breaches may be made.

In a large enterprise, you may end up with hundreds, thousands or even tens of thousands of authorization rules. The management of authorization rules in a large enterprise can be extremely complicated, time and resource consuming if not properly planned for.

Successfully Managing a WSSO Project

What are the business and security processes for agreeing to rule changes? How do you create the rules and test them in your test and pre-prod environments? How do you move the rules from test towards production? How much time and effort is involved per authorization rule change? What's the quality control? How do you search and find rules when you have a problem? How easy is it to find a specific authorization rule amongst hundreds or thousands? How do you know which authorization rules pertain to a specific set of web resources or applications?

Do you need to manage some form of delegation of who gets to see view or edit authorization rules? What's the business process for this? What's the management process for this? How easy is it to implement and maintain? Can your toolkit do this?

How are you going to handle protecting applications running on virtual servers? What are your strategies for protecting the numerous ways a URL can be specified and called? Who creates the strategy for this? Who signs off on this strategy within the enterprise? How do you educate the application owners about the various ways to lock down a URL? How is it tested before it's implemented? How are you going to enforce this?

The implementation and ongoing management of authorization is where the pedal hits the metal for WSSO. There are a lot of potential landmines when implementing new authorization strategies.

All too often you'll find out you probably assumed the WSSO product would handle it only to find out the hard way it either doesn't or, it's going to cost unplanned for money, time and resources to solve. This may be a result of not understanding your own processes as well as challenges with the selected toolkit.

You need to understand your own authorization processes in great depth, get beyond the sales spiels and dive to the detail within the WSSO products. Caveat emptor!

SESSION MANAGEMENT

Setting up WSSO authentication and authorization systems also requires attention and planning to session management. In large enterprise wide WSSO deployments, this can be complicated to set up and maintain.

Each application you integrate into WSSO is going to have its own session management conditions based on risk for timeouts and logouts. Your job is to figure out enterprise standards, if there are any, then decide how to provide exceptions to the standards for applications where it's warranted such that the protection matches the risk. Getting agreement at the enterprise level may take longer than you think.

Then you need to examine the requirements for each application as they are integrated into WSSO. What are their idle session timeouts? What is the application's maximum session timeouts? What are the user and application logout procedures?

With this in hand, the key question to ask your team and the vendors is how are you going to implement and maintain this? When it comes to applications such as portals, reverse proxies and others, you may find integrating their session management requirements into WSSO isn't always as straightforward as you may have thought. They may not be well setup to integrate their session management with WSSO. This may require lots more planning, coding, workarounds, time and expense you didn't anticipate.

Equally important is how you test for security integrity once you've set everything up. Do you have plans to try and break your own system? Can you find ways to spoof session management? Have you put the vendor's claims about their software to real life tests? What are your ongoing security testing programs?

The claims you're making to senior management need to be based on fact and not assumption. It won't do any good after a security breach to find out a basic hole existed from your implementation which testing could have taken care of.

Oftentimes, after extensive testing, the remaining weaknesses will be the process for updating all the components and notifying the WSSO system administrators of changes. Make sure your testing takes into account the underlying business and technical support processes. See if you can take advantage of your own processes to break your system.

AUDITING

While many people consider auditing almost as an afterthought, this is an area to be strongly thought out for reasons of legality, security and/or managing service level agreements. While many applications will have their own in-depth security (e.g. ERP), it may be very important to have an end-to-end audit view of all applications touched by a user during a WSSO session.

In the “old days” few users touched many systems. Today, a user’s session may include touching many systems frequently. You, your security and legal staff need to determine what acceptable risk is for auditing.

In general, it makes sense to have a high level overview of what applications a user touched during a session and when they did it. This kind of information has relatively low data volume. This should be supplemented with increased auditing on an application-by-application basis.

You need to determine which HTTP actions you’re going to monitor as a result of the risk assessment. For example, you may only want to audit only one of several possible HTTP actions. Then you need to create audit rules for the application, which take precedence over the more general enterprise audit standards.

Next you need to do some performance modeling to determine the impact which collecting the audit files will make on your system. The files may grow rapidly if there is much detailed tracking involved. There needs to be strong business and technical processes built into your WSSO system administration to monitor for this or, you may unwillingly bring your own system to its knees as the files sop up server performance and disk space.

For business, technical or legal reasons, enterprise service level agreements may require portions of the audit data to be intertwined with other traffic occurring between systems and users e.g. FTP audit data etc. You need to find out early on what information is required from WSSO and then determine how you will get portions of your WSSO system audit data intertwined with the other data. This could be complicated, expensive and time consuming.

Your WSSO administrators must be well trained to read and configure the audit logs coming from the WSSO servers and potentially the web and application servers the security agents are sitting on top of. When all hell breaks loose and your system is going down or gone down, you better have the team ready with an arsenal of tools and processes to quickly troubleshoot, diagnose and remedy the situation.

Some WSSO vendor’s audit logs are cryptic or lacking in detail. There may be inadequate information in the vendor’s support materials to really understand the nuances of the system. This is critical in a highly tense situation.

Do NOT let the consultants walk out the door without your staff knowing how to handle these situations. Demand excellent audit training and support material from the project consultants and the vendor.

Knowing the WSSO audit log system is a good starting point but not enough. In an n-tier environment, the problem may lie in a process that's gone screwy in one of the web, application, directory, portal or reverse proxy servers. WSSO is the system where many of these other systems have a common touch point. In a potential system failure, people may start pointing fingers at each other with you in the middle without adequate data to back them up.

Your job is to minimize this risk. There should be clear business processes, meetings and training between the infrastructure parties that lead to inter-testing with sharing of audit logs under approved processes. While this may take a lot of patience and effort to get people to agree, prioritize and do, ask yourself how much the enterprise will lose for every minute you're down? That should help you garnish the support from the executive suite to get everybody on the same page.

In the early days of the cold war, the distant early warning line (or "DEW" line) was deployed into North America to give advance warning of incoming threats. You need to be thinking the same thoughts with respect to your WSSO system. Unfortunately, WSSO vendors and consultants often view auditing as a measurement of something that happened rather than monitoring something that's happening.

You must direct your staff, vendors and consultants to give you the tools such that you can be alerted to a developing problem before it escalates into a system failure. Being able to parse and understand the audit logs in real time is a key part of this. Many vendors' toolkits are lacking in this regard. You're likely to end up writing your own code.

You WSSO system administrators need a WSSO management console that can trigger alarms from audit log information in near real time. If an authentication or authorization server comes under attack or, the WSSO system performance starts to slow due to some aberration or, something like authentication failures starts to dramatically climb, you'd like to know of it a few seconds later and have the expertise and processes to be rectifying it before it becomes a disaster.

These are just a few of the many audit considerations you need to be aware of and plan for. Don't wait until it's too late to find out the hard way that you should have paid more attention to your audit requirements.

INFRASTRUCTURE MAINTENANCE

Following on from the audit discussion of understanding audit files in near real time, you need to focus on WSSO infrastructure maintenance. Earlier on in this paper, the question was posed as to the consequences from WSSO not working. Your job is to ensure that the infrastructure is always up. It's my experience that this is an area requiring extensive planning, documentation, testing and one that too often escapes adequate management attention in the beginning.

It starts off with the physical infrastructure planning for the production version of the authentication and authorization servers. Well before you've ordered your hardware, very careful thought needs to be done in advance for capacity planning, performance, failover and disaster recovery.

What are the short, mid and long-term capacity requirements? What are the assumptions you've made for the authentication, authorization and auditing models? Are your authorization models based on page views or on HTTP requests? Have you checked with the WSSO vendor to see when a trip to the policy server is required? What are the caching conditions you're using in your model? Does your model take into account overhead from encrypting and decrypting traffic? How can you justify your assumptions?

Beware the vendors' published performance numbers. These are for competitive sales analysis, not necessarily mimicking the conditions on your hardware, your users, your applications, your network and your WSSO settings you decide upon. How are you going to test for real world use on your systems?

How are you going to measure performance? What kind of system checks are you going to make on your WSSO servers at the hardware and WSSO application level? Are these automated? Who's setting this up? How does it intertwine with the many back-end enterprise support systems? What are the escalation procedures?

If you're using more than one network OS and various hardware platforms running web, app and portal servers that each in turn have a WSSO security agent running on them, the infrastructure support of this can be extremely complicated and time consuming to setup, test, maintain and troubleshoot. Do you have the right people on your team to do this? What tools does the vendor give you to integrate with the support systems? What additional tools do you require?

What's your failover strategy? How do you do routine maintenance? Who gets notified? What's your quality assurance strategy for all this?

You must have:

- Extensive documentation
- Extensive scripts to automate as much as possible of the maintenance processes
- Extremely knowledgeable staff who understand the nuances of supporting an n-tier model
- Complete toolkits to quickly respond to a problem or disaster in the making, rather than after the fact
- Extensive testing to prove that all the aforementioned works in real life.

Successfully Managing a WSSO Project

What is your strategy for security hardening all WSSO components? Do you have a plan, documentation and testing for things like internal and external attacks on the WSSO system?

What is your migration plans for the WSSO software? What kind of testing, toolkits, documentation and procedures are required for migrating out of a test environment, to pre-production and production? Do you have enough hardware, budgets and people to do set up these environments and then manage it? If you are interfacing with multiple types of web, app and portal servers, running on different NOS's, creating the test environments for these with WSSO may not be trivial.

Do NOT let the consultants and vendors set the WSSO system and infrastructure up with few of your own staff understanding why things were done the way they were. This is folly. There are so many nuances unique to each n-tier system than you cannot afford to have the knowledge of these nuances that affected WSSO design, walk out the door with the consultants.

You cannot afford long fix times to your mission critical WSSO system due to a lack of knowledge or understanding of your staff. Make sure the consultants, project manager and your staff adopt excellent documentation standards from the outset of your project. Ensure a solid knowledge transfer occurs at all steps along the way or, you and your enterprise may pay an unnecessary stiff price for partial or full WSSO system failure or security breaches.

CONCLUSION

Managing a web based SSO project in a large enterprise is not trivial. The n-tier environments of most large enterprises creates a wide set of variables and nuances that must be addressed in WSSO planning. All too often enterprise managers find out this reality the hard way.

It's easy to let the consultants and technical staff hunker down into their cubicles and meeting rooms, only to find out a fair way into the project that some of the pieces aren't properly stitched together and/or the project won't be delivered on time and budget. There are a number of warning signs in advance to help steer you out of danger.

Watch out for extensive process but not enough diving to the details. Expect and demand tough detailed upfront questioning of all the systems WSSO will touch before the project gets down to the doing stage.

Watch for and demand extensive testing plans that are well staged throughout the project. Don't wait several months to find out you had major problems that could have been detected earlier. Continuous testing will also help verify claims that your identity, authentication and authorization data stores, networks and other infrastructure owners make as to their systems integrity and performance.

Look and demand for very detailed system maintenance management plans. Making the WSSO system work with your infrastructure plumbing and their alarm and response systems can be complicated. It's also mission critical.

Make your team demonstrate to you screen by screen how either they or other managers are going to manage hundreds or thousands of rules. Visualize in your mind your systems a few years down the road with the hundred or thousands of rules. Picture the change requests. Figure out the delegation model. Relate this picture in your head to the workflows your team is showing you on the pilot screens today.

Don't let the WSSO team and consultants buffer your enterprise manager responsible for WSSO from the details. You need a manager who can understand the details. WSSO is a detailed application infrastructure-plumbing project. The devil is in the details. If your WSSO manager stays up at only the high process and vision level, you may end up in trouble when reality sets in.

Choose your consultants and staff wisely. Too many people say they can do WSSO but have not lived a complicated project. It's a lot more than some WSSO toolkits and consulting project management processes. Beware the slick answers. Demand and expect details with realistic appraisals of the situations.

If you do your homework well and properly plan for web based SSO, you're in good shape. When properly implemented, WSSO can help standardize identity, web and application security, enhance ease of use and position the enterprise for rapid, secure integration with business partners' and customers' systems.

USEFUL REFERENCES

[“Single Sign On Underneath The Hood” - What Senior Managers Need To Know - \(pdf\)](#).

This paper covers the areas of planning for a Web Single Sign On project that senior managers should be aware of before starting a project.

[Five Minute CEO/CIO Briefing on Web Security- \(pdf\)](#) or [\(html\)](#)

The first in a series of papers covering web security by Guy Huntington, HVL. This briefing covers seven questions CEO/CIO's should be able to answer to understand web security.

[Reducing Total Cost of Ownership in Web Security- Six Points to Saving Money \(pdf\)](#) or [\(html\)](#)

The second in a series of briefings on web security by Guy Huntington, HVL. This briefing is designed to quickly educate senior managers where to look for cost savings in web security deployments.

[Web Security PowerPoint Presentations \(www.hvl.net/ebusiness.htm\)](#):

Put together by Guy Huntington, [HVL](#) and Derek Small, [Nulli Secundus](#) from their web security, directory and ERP experiences, these presentations cover general and specific issues in web security, identity management, single sign on, LDAP directories, PeopleSoft LDAP integrations and PeopleSoft Oblix integrations.

[“Secrets and Lies: Digital Security in a Networked World”](#):

Bruce Schneier, CTO of Counterpane and a noted cryptography expert wrote an excellent book covering the issues of digital security. It's a must read for IT managers.

[“SAML” \(Security Assertion Markup Language\)](#):

This is the emerging standard for authentication and authorization. It's being put together by OASIS, the Organization for the Advancement of Structured Information Standards. OASIS is a nonprofit, international consortium that creates interoperable industry specifications based on public standards such as XML and SGML, as well as others that are related to structured information processing.

ABOUT THE AUTHOR

Guy Huntington, President of [HVL](#), recently was the project manager for Boeing's global single sign on initiative. He specializes in leading enterprise re-engineering projects including B2B's, B2C's and intranets integrating web security and LDAP directory infrastructure into the many disparate systems within each enterprise including data warehouses, ERP's, etc. Guy has lead many projects over the years, authored numerous papers, on-line presentations, written an electronic book and given many speeches on the use of the web in business. Guy can be contacted at 604-921-6797, guy@hvl.net or www.hvl.net.