GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Implementing Single Sign-On – Imprivata OneSign™

GSEC Gold Certification

Author: Robert S Turner, rspt@charter.net

Adviser: Joey Niem

Accepted: [Insert Date]

© SANS Institute 2007, Author retains full rights.

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

© SANS Institute 2007, Author retains full rights.

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

1. Introduction

During my SANS GSEC course, we spent very little time discussing single sign-on (SSO) technology. At the time, I was working with a SSO technology, and it occurred to me that this would be a great topic to bring to other course participants.

For the uninitiated, single-sign on is more of a concept than a specific technology. In this case, we are using a combination of hardware and software to act as a password vault. One user ID and password combination is used to access the vault, which stores the rest of the user's credentials for various applications.

In our hospital setting, a single user can encounter numerous applications and authentication requirements, each with different expiration intervals and complexity rules. I believe SSO can significantly help users with the need to remember multiple, complex credential sets.

In this paper, I will focus on the implementing SSO with the Imprivata OneSign™ Appliance. The Imprivata website boasts, "OneSign™ Single Sign-On quickly and effectively solves password management and user access issues. OneSign™ single sign-on enables ALL applications – legacy, client/server, and web - without requiring any custom scripting, changes to existing directories, or inconvenient end-user workflow changes. OneSign™ Single Sign-On dramatically lowers Help Desk costs associated with forgotten password resets, increases user productivity and satisfaction, strengthens password security, and supports regulatory compliance initiatives."[1]

Making the claim "ALL applications" seems quite bold. Can OneSign™ live up to that assertion? With this paper, you will be able to examine their technology and my real-world experiences, rather than just vendor hype.

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

2. Project Scope

I developed project plan for this test which included the following scope criteria:

Scope

- Configure all business critical apps for SSO
- Integration with existing infrastructure
- Implementation of failover and redundancy
- Client based disaster recovery mode
- Implement new password complexity standards
- Introduce multi-factor authentication
- Reduce login delay where possible
- Comply with HIPAA standards
- Simplify passwords for users
- Reduce calls to Help Desk

The remainder of this paper will examine how OneSign™ performed for each objective.

3. Configure business critical applications

The primary stage gate for project success turned out to be the most exciting item within our test: Ease of application setup. Of all of technology within the OneSign™ appliance, the Application Profile Generator™ (APG) is the one that impressed me the most.

To create an SSO enabled application, we used the APG interface from the administrative web console. To a significant extent, the APG was able to identify all titles, fields and action buttons available on any login dialog. For Windows-based applications, all that was necessary was to continue to follow the numbered steps, and then save the application profile. The

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

code for each application is written by the APG in Extensible Markup Language (XML) and stored o the system.

In some cases, it was necessary to modify the XML code by inserting keystrokes, such as {TAB} to move from field to field or {ENTER} to submit the credentials, but once you have worked within the APG, that becomes trivial. For applications that launched slowly, we needed write custom codes to add a wait time value to allow the login dialog to load completely before submitting the credentials. This was a simple trial-and-error process that took just minutes to work through.

Finally, we did have two applications that could not be properly recognized by the APG. In these cases, we created custom profiles that could recognize some of properties of each login dialog box. When the application is launched and subsequently recognized, a specially designed intermediary login form, called an Explicit Credential Capture Dialog, pops up to capture the credentials and feed them to the application.

Our applications for the pilot included HTML and Java browser based interfaces, terminal sessions, native windows and Citrix® based windows applications. Within two weeks of the beginning of the pilot, we had 13 applications enabled and working to a high degree of accuracy. Of these applications, our lone problem was with the modal state of some applications. If the login dialog lost focus prior to application recognition, the SSO process would not automatically recover once focus was regained. This most often occurred when our trial staff would open multiple SSO enabled applications at in rapid succession. We solved this issue the old fashioned way; patience. We asked users to wait for each application to login before starting a second application.

Objective grade - Pass

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

4. Integration with the current environment

Our existing environment met with all posted requirements and consisted of network LDAP infrastructures of two clustered Novell eDirectory Servers as well as a Windows 2000 Active Directory Domain. The workstations used were HP/Compaq Intel-based PC's running Windows XP Professional, SP1. Since our primary authentication is with eDirectory, we configured the OneSign™ appliance to work with that directory from the outset. The full list of requirements can be downloaded from: http://www.imprivata.com/content846.html.

We connected the appliance to the network and provided IP address and DNS information to the appliance with front console. From that point we were able to log into the web-based administrative console to complete the configuration. Using the Security Policy Configuration we were able to quickly select and import the Novell user database of just under 2000 within a few minutes. The Security Policy includes many options, including which multifactor authentication modes can be used within the policy. More than one policy can be defined to suit the security needs of the user or organization.

After testing with eDirectory, I added the Active Directory LDAP catalog to the mix. This enabled me to synchronize AD passwords with the eDirectory domain passwords. In other words, if an end user changed their eDirectory password, the matching Active Directory account would be updated as well, using the OneSign™ appliance as a password synchronization appliance. This works only when the login name for a single account is the same for both domains.

Later in the pilot, I set Active Directory as the primary directory for the Security Policy and completely removed the eDirectory catalog. When this happens, all users will need to

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

be re-enabled for SSO and subsequently re-enroll with any multifactor devices you may be using as well store application passwords. Both directories performed equally as well within our test environment.

There was one infrastructure issue that I needed extensive support with was client installation directly from the appliance. When setting up a new user, you can automate an email notice which includes a link for the user to download the appropriate client. This helps to automate the install process. Unfortunately, my success rate was only about 50% for using this process to install the client. As a work-around, I was able to download the .msi file directly and install it using the Control Panel -> Add/Remove Applications Windows applet to control the process. This worked every time. We reported the issue to Imprivata and continued the pilot.

Another goal within our infrastructure was to have client mode compatibility for our diverse environment. There are three client modes for connecting, each with different advantages.

The single user mode is used on a pc with a standard configuration that would have only one user working from it. This is the most common client and is the default installation selection. Visually, this adds an extra component onto the current GINA (Graphical Identification and Authentication) user interface. Functionally, it logs a user into the network and the appliance in rapid succession.

Shared User Mode – This was a very useful client mode within our environment. We have several areas where the computers are shared. Depending on the account configuration and access rules this can cause issues with the HIPAA requirement to use unique logins. 45 CFR § 164.310 states, “Unique user identification (Required). Assign a unique name and/or number for identifying and tracking user identity.”[2]

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

The shared mode installation looks the same as the single user, as it adds a component to the existing GINA, however there is a difference when the user logs on. The first login will authenticate to the network only. Immediately following, an additional pop-up dialog prompts the user for the SSO credentials. The beauty of this method is that the network authentication could be setup to give access to shared Novell Delivered Applications that everyone would need, but require SSO authentication to use SSO applications that contain Protected Health Information (PHI). This allowed rapid computer sharing because a change in user only required a log off and log on to the SSO appliance – the shared network user never changed, thus eliminating the time needed to run login scripts, map drives and establish windows profile settings.

Finally, there is a specific installation for use with Citrix®. The client is installed on the server and when the user authenticates to Windows™ Terminal Services, the OneSign™ agent is called and authenticates to the OneSign™ appliance. When using the Citrix® client mode, the recognition of the application was different than that of single client mode applications, prompting a need to create second set of application profiles. Based on maintaining the ability flexibility of using thin client applications, we feel it was worth the effort to create the redundant application profiles.

Objective Grade - Pass

5. Failover and Redundancy

The implementation is easily setup to address potential failure. A second, hot backup appliance is configured in failover mode and connected to the primary appliance via a cat5e

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

or cat6 cross-over cable. If the primary fails, the secondary takes over all processing automatically.

For redundancy, an encrypted backup configuration file can be made and stored in a location that is backed up during your normal tape backup strategy. The resulting file can be loaded from file or tape to a cold appliance to have the on-line authentication mode restored within minutes.

During the downtime for failover or restore, clients can still use SSO with cached credentials, which is discussed in the next section.

Objective Grade - Pass

6. Client based disaster recovery mode

One of the main concerns from our I.S. department was, “What if the appliance goes down; People won’t be able to login?” This is handled in two ways. First, just because an application is SSO enabled, does not mean that you cannot access it by logging in manually. Second, the appliance can be configured to stored cached credentials on each client machine. The only pre-requisite is that an SSO enabled user had previously logged into the PC that they wanted to use. If so, all of their credentials would be stored within their Windows™ profile and updated during the course of normal use. For increased security, this cache is 3DES encrypted and can be set to expire within a specified time period. This would come in handy in the event of a lost computer or if employee who had been terminated attempted to access the computer.

Objective - Pass

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

7. Implement new password standards

There is a nice side benefit to implementing single sign-on when there is resistance to strong complexity rules for passwords within your organization. In Perfect Passwords: Selection, Protection and Authentication, the author states that passwords have long been the most pervasive security weakness in any system. "System administrators complain that users select weak passwords and end users complain that IT departments set draconian password policies that make their lives difficult. One policy users hate most is changing their passwords every two or three months. Users typically respond to these policies with predictable patterns, for example adding a number to the end of the password and incrementing it each with each subsequent password change."[3] For this project, we simply delivered an upfront expectation: We are going to REDUCE the number of user ID and password combinations you need to remember and, in exchange, INCREASE the complexity of the one password you do need to remember. This is one of the greatest benefits of SSO: "For password-based authentication, this means only one password to remember and update, and one set of password rules."[4]

We were able to configure OneSign™ with a wide variety of complexity rules including character sets, expiration time periods and password recycling options. The complexity rules are established per Security Profile, so users or groups of users can be treated differently, depending on their specific needs.

Objective Grade – Pass

8. Implement multi-factor authentication

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

While SSO does not require a multifactor login implementation, it was an important item for us. We didn't want to force use of multifactor, instead, we wanted to make it available as an alternate, strong means of authentication. With every installation of a multifactor authentication device, we included the means to bypass that method and login with a user name and password. This is critical in a clinical environment where excessive login delays, caused by system issues, would be unacceptable. With that in mind, I spent a significant amount of time working with the multifactor technologies.

Our unit shipped with a pair of Upek TouchChip® TCRU1C biometric USB fingerprint readers (fig. 1), which were used in the pilot.

Figure 1 – Upek Sensor

Using Upek fingerprint biometrics and Xyloc proximity cards with OneSign™ is simple plug-and-play installation. The support is built right into the appliance and the hardware drivers are included with the client installation. We'll discuss Xyloc cards later in this section.

To use biometrics with OneSign, the user will first need to enroll. This would happen automatically if the device is already installed on the computer. Users have a choice of recording between one and ten fingerprints. I highly recommend at least three impressions, with at least one on each hand. This will give alternate sources of authentication in case of a print alteration, such as a cut or dressing on the finger. If an alternate images has not been stored or in the case of multiple failures, users can always simply login by typing their ID and password.

Our fingerprint enrollment went very well with two noted exceptions. First, when you tell a user to press their finger onto the reader, sometimes they press down very hard. Sensors

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

are designed to account for some distortion when recording details, but with excessive pressure distortion can cause errors in obtaining the initial scan.

In fingerprint matching tests, researchers (Jain & Pankanti) discovered that "variations in the area of finger being imaged and its pressure on the sensing device affect the number of genuine minutiae captured and introduce displacement of the minutiae from their 'true' locations due to elastic distortion of the fingerprint skin. Such elastic distortions and feature extraction artifacts account for minutiae matching errors even after the prints are in the best possible alignment."[5]

In our other cases, the distorted prints were accepted as accurate and converted to a hash value to be stored on the appliance. This yielded Type I (false negative) errors for subsequent authentication attempts.

These errors could be easily corrected by deleting the stored fingerprint hash and re-enrolling. During the process, I got in the habit of instructing users to apply the same amount of pressure as if they were picking up a pencil. This technique all but eliminated these types of errors.

The second set of enrollment hurdles we encountered are issues referred to as non-uniform contact and irreproducible contact errors. These two contact issues can result with elderly users or users with skin injuries. In our case, the primary problem was with older enrollees.

"Non-uniform contact can result when the presented fingerprint is too dry or too wet, and irreproducible contact occurs when the fingerprint ridges are semi-permanently or permanently changed due to manual labor, injuries, disease, scars or other circumstances such as loose skin. These two contact issues can result when an elderly user (62 and older)

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

presents their fingerprint to the fingerprint device. As individuals age, their skin becomes dryer, sags from the loss of collagen, and becomes thinner and loses fat due to the loss of elastin fibers, which decreases the firmness of the skin, and is likely to have incurred semi-permanent or permanent damage over the life of the individual."[6]

For users who encounter these issues, fingerprint biometric authentication should not be used.

Since our users are all issued HID Prox II (fig. 2) cards as part of our ID card system for physical entry, I also tested a pcProx-232 reader.

Figure 2 – HID Prox Badge and Reader

This required a download from http://www.rfideas.com/html/downloads.html, but after a brief configuration, the client was ready for use. The first login required that I type my credentials to login. I was then prompted to swipe my Prox card to assign its ID to my user ID on the appliance. From that point I was able to authenticate using either the Prox card alone or in conjunction with a password.

On the OneSign™ appliance, you can configure a card to work alone, with a password, with a biometric device or with a token generator. In the end, we chose to go with a card for identification plus fingerprint for authentication, even though it did introduce more desktop peripherals. The reasoning behind this decision is simple: It is highly unlikely a user will give away a finger to share their authentication. While studying two-factor authentication Julian Curmi found that, "No matter what type of two-factor authentication model is used, the organisation [sic] should be sensitive to the fact that proper implementation is key to the reliability and security of the

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

system. For example, a poorly implemented two-factor system may be less secure than a properly implemented single-factor system because of weak organisational [sic] policy, procedures or standards. This is so, because the human element is the weakest link in any security application or system."[7] Simply put, a smart card and password is just as easy to hand off as a password alone.

We also tested Xyloc cards and readers (fig. 3), which were a big hit with our user community. The big difference between the HID Prox and the Xyloc card is that the Prox card is not an active transmitter and therefore has to be swiped against or near the reader. Xyloc on the other hand, can be configured to read the card out as far as 50 feet away from the reader eliminating the wait time for the user to make the card swipe. (www.ensuretech.com, 2006)

Figure 3 – Xyloc Transmitter and Receiver

Among other features, I liked the Xyloc card because it required no user interaction to lock the workstation. If user left their work area, the reader would no longer sense the badge and automatically lock the workstation within a defined number of seconds. When the user returned the badge would be read again and submit only the User ID portion of the credential set. To complete the login, the user must submit fingerprint or password authentication.

Objective Grade – Pass

9. Reduced login delay

Despite all of the previously listed accomplishments, the project would not be a success if we introduced delay into the

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

login process. I've never worked anywhere where the users didn't already complain about slow networks.

In our testing, login times were primarily reduced by using the previously described shared mode client. That alone shaved between 40 and 100 seconds off the login wait. Additionally, for hunt-and-peck style typists, using a badge and biometric combination can reduce login time by a few seconds.

Application login times saved only a second or two, but over time those seconds can add up, even if only in user perception. "Using SSO to simplify authentication expedites users' access to applications and networked resources. Just like home computing, users double-click what they want and get it a second later. Immediate gratification may sound trivial, but those microseconds add up; even the slightest of hang times can frustrate a user."[8]

Finally, we could not add time by waiting for authentication to the appliance. Because we did add an intermediary device into the authentication chain, there could have easily been some induced delay. According to research done by Imprivata, the sum of packet sizes transferred via HTTPS from client to appliance and back is 24kb, making the network performance hit next to nothing.

Objective Grade – Pass

10. Comply with HIPAA standards

Another OneSign™ benefit is the ability to force complexity rules within each application as well as the domain. In the health care environment, increased password complexity, password change time restrictions, password recycling policy all have roots within HIPAA Security Standards and make the enhancement

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

of password policy a very strong selling point. In accordance with HIPAA Standards, we must either have a procedure for, or document why we cannot have, secure passwords; "Password management (Addressable). Procedures for creating, changing, and safeguarding passwords."[9]

Application security is enhanced by allowing OneSign to intercept password change prompts from each SSO enabled application. Change Password screens can be created within each Application Profile using the APG. If the screens are configured properly, the appliance will respond to password change requests on behalf of the user, updating the password within the application and logging in with the new credentials in a matter of seconds.

Another regulatory benefit gained from implementing OneSign™ is auditing capability through the Reports page. Several of our older applications, as well as those that were home grown by the organization, did not have the auditing features necessary to comply with the auditing standard. Of course, log-in monitoring is specifically addressed within HIPAA; "Log-in monitoring (Addressable). Procedures for monitoring log-in attempts and reporting discrepancies."[10] OneSign™ enhanced our ability to monitor those applications by tracking user related events such as login, enrollment, lockouts and other activities and presenting them in a very readable, customizable report.

Objective Grade – Pass

11. Simplify Passwords for Users

In addition to the goal of increasing password complexity, we also wanted to simplify the overall authentication process for our users. Although this might sound contradictory, we

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

surely must realize that when users have too many passwords to remember problems are inherent. “Of those organizations that do have strict enough password policies, many users simply resort to writing their passwords down on sticky notes and slap it on their monitor because they're too difficult to remember. The reality is that we have now reached an era where commodity computing power has exceeded the average human's ability (or willingness) to remember sufficiently complex passwords.”[11]

OneSign allowed us to go from credential sets of 9 and 10 to just one. In work areas where multifactor authentication was being used, we only needed to remember the user ID portion of our user ID/password pairing. To make remembering the complex passwords easier, I recommended that our trial staff use a mnemonic for a password. For example, “Sigtg4mwin!”. This is a quote from one of my favorite characters in one of my favorite movies “...so I got that goin’ for me, which is nice”.[12] Obviously, this is now a password that I will never recycle.

In the end, any SSO solution has the native potential to simplify the password experience. Imprivata OneSign™ lived up to that requirement in grand fashion.

Objective Grade – Pass

12. Reduce calls to Help Desk

The final criteria to meet was to reduce password related calls to the help desk. According to Gartner, “About 30 percent of all helpdesk calls require a password reset, at cost of $25 to $50 per call.”[13] OneSign meets this challenge with what they call Self-Service Password Management.

Prior to using the self-service feature, users must enroll in this service by answering a series of challenge questions. For example a challenge question might be, “What is your pet’s

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

name"? The Security Profile then can be set to determine how many questions to ask and how many must be answered correctly to authenticate the user. If user forgets a password, they access an internally available website and answer the challenge-response series. Once the user is authenticated, they can choose to reset their primary password or request application passwords.

This OneSign™ feature allows for 24/7/365 access to passwords and password resets, without any interaction from the help desk. While our pilot did not last long enough to gather significant metrics, we made note that users did indeed log into the Self-Service web page and used its features.

Objective Grade – Pass

13. Conclusion

Many of the objectives that I have discussed here are not unique within Imprivata OneSign™ technologies. There are a wide variety of offerings for tools to administer user accounts. From password and rights management to accounts provisioning and physical access, the disparity and availability of SSO technology is formidable. I think what sets the Imprivata OneSign™ technology apart from the others is the ease at which it accomplishes these objectives. My testing met with superior success. I met all of my scope criteria, including enabling 13 business critical applications with little difficulty and introducing multifactor authentication, while encountering a great deal user satisfaction along the way.

Others have had a similar experience "OneSign™ truly allows easy implementation and deployment of Single Sign-On functionality, eliminates the need for back-end coding of

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

applications, replaces many logons with one centrally managed secure login and, most importantly, reduces costly password related calls to the Help Desk. OneSign™ is not an inexpensive product but the return on investment through eliminating downtime due to user lockouts is well worth it."[14]

Finally, I want to make sure that anyone who reads this paper understands that I am not affiliated with Imprivata in any way and neither I, nor my company is being compensated for this work. I understand that a paper written from this perspective can read like a glossy advertisement. The simple truth is that during the course of my proof-of-concept testing, I found that I really liked the product and thought I would share my experience with my peers.

© SANS Institute 2007, Author retains full rights.

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.

References

[1] Imprivata One Sign. Retrieved February 4, 2007, from Sign On and Network Authentication Solutions Retrieved Dec 7, 2006 from Imprivata Web site: http://www.imprivata.com/products/
[2] U.S. Government Printing Office. (2004). Health Insurance Portability and Accountability Act (HIPAA) of 1996 (45 CFR § 164.310 (a)(2)(i)) Retrieved December 20, 2006 from http://a257.g.akamaitech.net/7/257/2422/05dec20031700/edocket.access.gpo.gov/cfr_2003/octqtr/pdf/45cfr164.310.pdf
[3] Burnett, M (2005). Perfect Passwords. Rockland, MA: Syngress.
[4] Botzum, K (2001, Aug 14). Single Sign On - A Contrarian View. Retrieved February 4, 2007, from IBM Web site: http://www-128.ibm.com/developerworks/websphere/library/techarticles/0108_botzum/botzum.html
[5] Jain, A, & Pankanti, S Automated Fingerprint Identification and Imaging. Retrieved February 4, 2007, from http://www.research.ibm.com/ecvg/pubs/sharat-forensic.pdf.
[6] Elliott, Stephen J., Kukula, Eric P., & Sickler, Nathan C. 2004). The Challenges of the Environment and the Human / Biometric Device. 3. Retrieved Dec 10, 2006 from Web site: http://www.biotown.purdue.edu/proceedings/HBDIandEnvironmentCanada2004.pdf
[7] Curmi, Julian The Importance of a Two-Factor Authentication. Retrieved Dec 29, 2006, from http://www.speedyadverts.com/SATopics/html/information_security1.html
[8] Walsh, L (2004, Nov). The User Experience. Retrieved February 4, 2007, from Information Security Magazine Web site: http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss506_art1047,00.html
[9] U.S. Government Printing Office. (2004). Health Insurance Portability and Accountability Act (HIPAA) of 1996 (45 CFR § 164.308 (a)(5)(ii)(D) Retrieved December 20, 2006 from http://a257.g.akamaitech.net/7/257/2422/05dec20031700/edocket.access.gpo.gov/cfr_2003/octqtr/pdf/45cfr164.310.pdf
[10] U.S. Government Printing Office. (2004). Health Insurance Portability and Accountability Act (HIPAA) of 1996. 45 CFR § 164.308 (a)(5)(ii)(C) U.S. Government Printing Office. (2004). Health Insurance Portability and Accountability Act (HIPAA) of 1996 (45 CFR § 164.308 (a)(5)(ii)(D) Retrieved December 20,2006 from http://a257.g.akamaitech.net/7/257/2422/05dec20031700/edocket.access.gpo.gov/cfr_2003/octqtr/pdf/45cfr164.308.pdf
[11] Ou, G (2007, 1 11). Ultimate Wireless Security Guide. Retrieved February 4, 2007, from Tech Republic Web site: http://articles.techrepublic.com.com/5100-1035-6148551.html
[12] Kenney, D (Producer) Ramis, H (Director).(1980) CaddyShack [Motion Picture]Orion Pictures Warner Bros
[13] Wilson, T (2006, Dec 7). Oracle Spurs Single Sign-On Surge. Retrieved February 4, 2007, from Dark Reading Web site: http://www.darkreading.com/document.asp?doc_id=112382
[14] Kvitka, Andre. Review: Simple single sign-On. Federal Computer Week, March 04,2005. Retrieved November 28, 2006 from Web site: http://www.mwavemedia.com/fcw.pdf

© SANS Institute 2007, As part of the Information Security Reading Room Author retains full rights.