Global Information Assurance Certification Paper

Controlling Remote Access for Vendor Support

Mark A. Cooper
GIAC Security Essentials Certification (GSEC)
Practical Version 1.4b Option 2
May 2003

Controlling Remote Access for Vendor Support

Abstract

Controlling and auditing remote vendor access to banking systems is not only a fiduciary responsibility of a financial institution but also an indicator of the safety and soundness of support and delivery systems. Without an internal procedure for controlling vendor remote access, the vendor retains a greater level of responsibility for the security and integrity of the systems than the customer is willing to entrust.

Through a process of identifying and equipping internal users with a set of Active Directory controlled vendor login accounts, an auditable trail of both internal user actions and vendor support activities was created to enrich the accountability of our vendor and our internal personnel.

While remote access capabilities are an inherent weakness in the overall security of a network, implementing remote access controls that include the ability to facilitate auditing mitigate the risks associated.

Controlling Remote Access for Vendor Support

Introduction to Case and Background

Financial institutions face a regulatory environment unlike most industries. Concerns of safety and soundness of financial condition and support and delivery operations drive federal level (and state level in some instances) oversight to what is best described as a fevered frenzy. All this to the eventual and necessary soothing of the psyche of the banking customer who constantly needs to be assured that the money they have on deposit is properly accounted for, secure, and immediately available.

For the financial institution to satisfy the regulators charged with evaluating the implementation of guidance directed policies and practices, constant assessment of risk and remediation of the risks identified is necessary. Chief among the risks identified is the risk of uncontrolled, unauthorized, and/or unmonitored access to the systems housing the deposit records of the financial institution including access by the software system vendor.

In financial institutions that have chosen to internalize the processing of their customer transactions, vendor support of the systems is for all practical purposes a requirement. Program customization, parameter customization, debugging of discovered anomalies, and general operational support all necessitate the vendor being able to access the system. Developing a method of controlling vendor access balanced against the efficiency needs of the vendor and the operational needs of the financial institution becomes a delicate dance.

Financial institution core processing platforms are accounting systems on steroids. The majority of in-house core platforms are run as batch updated account balance and interest accrual systems that accumulate adjusting transactions throughout the business day through very tightly controlled double-entry controls. Only after an end of day sorting and balancing of all posted transactions does the actual update process begin. Strict adherence to accounting principles, to regulatory guidance, and to the account handling agreements and disclosures executed between the bank and the account holder is required.

Supporting and managing the actual programmatic steps required to accomplish this task usually exceeds the internal knowledge capacity of the organization. The burden for this support is therefore contracted to the vendor that developed and installed the system in as much as the intimate knowledge of the programming of the system, the ability to make parameter changes that affect the processing, and direct access to the transaction databases to bypass the double-entry controls exists only in the hands of the support personnel in the vendor

organization. This leaves the financial institution with the task of operating the system, validating the data, generating customer statements and management reports, and most importantly, managing the vendor.

To evaluate our position and develop a set of requirements for managing vendor access to our systems, the following documents were referenced:

> Regulatory examination by governmental organizations (such as the FDIC, OTS, and FSLIC) is based on criteria established by the Federal Financial Institution Examination Council (FFIEC) and published in the FFIEC IT Examination Handbook. The current handbook was adopted in 1996 but is currently being revised through a series of booklets that will eventually replace all sections of the handbook. The first booklet in the series was released in December 2002 and deals with "Information Security".
>
> The IT Governance Institute™ sponsored by the Information Systems Audit and Control Association (ISACA) has designed and created a set of publications through their COBIT Steering Committee which defines a system for applying "Control Objectives for Information and related Technology". This system provides a framework through which "a generally applicable and accepted standard for good IT security and control practices to support management's needs in determining and monitoring the appropriate level of IT security and control" can be established and validated.

Based on the general guidance from these sources, a relatively simple method was used to guide our procedure definition: the questions Who?, What?, When?, Where?, Why?, and How? were asked to determine the basis for our design and to measure the effectiveness of our design:

Who?	-	Who was in our system? Who authorized them?
What?	-	What were they doing while they were in the system? What were they supposed to be doing?
When?	-	When was the authorization given for access? When did they access the system? When did they finish?
Where?	-	Where on our systems did they access? Where did they access the system from?
Why?	-	Why was the access necessary?
How?	-	How was access to the system affected?

Before

Prior to implementation of our defined system, vendor access to the systems was facilitated through the use of a common username to log on to our network by way of a dialup remote access service. The vendor support staff numbers in excess of 30 individuals who might need to be able to gain remote access at any given time due to varying functions in the support group and on-call rotations.

A common username was created on the network by the vendor at the time the systems were initially installed and a non-expiring password was assigned to the account. This system was put into place because

a) it establish a standard method for their support staff to access the system thereby reducing the complexity of their support infrastructure,

b) it required limited involvement of our internal operations staff or internal information technology staff as far as providing any technical assistance to the vendor when the vendor needed access, and

c) it ensured that the vendor support personnel would be able to access the systems as they deemed it necessary.

While this was good for the vendor it was not necessarily within our best interest. When we asked ourselves the following questions from a vendor actions audit perspective, the answers were not within our defined level of risk acceptance:

Who?

1. Having the vendor use a common username prevented us from being able to determine which of the vendor support personnel was currently in the system or had been in the system. They could not be held accountable for their activities.

2. A non-expiring password on the vendor support account prevented us from being assured that the account was not being used by anyone other than currently authorized vendor support personnel. It was conceivable that through employee turnover at the vendor, that an ex-employee (or other non-authorized person) could gain access to our system.

3. With open vendor access to the systems, we were unable to manage who within our organization was authorized to initiate a support request to the vendor and ultimately affect the operation and function of the system.

What?

4. Excess authority of the vendor support account enabled them to access other resources on the network that were beyond their needs to be able to support the systems.

When?

5. Not having any imposed time of day restrictions on access prevented us from knowing when the vendor support personnel were accessing the system.

Where?

6. While our event logs showed the login activity of the vendor support personnel which included the name of the computer being used to launch the access, specific tracking information about which systems were being accessed was lacking.

Why?

7. No auditable method for correlating our requests for vendor support to the vendor accessing the system was available. They could access the system without being requested and perform functions of which we were not specifically aware.

How?

8. More than one point of access existed through which the vendor could gain access. While each access point created log entries when they were used, determining how the access was gained required checking in several places for the records.

As our organization has grown and our familiarity with the supported system has increased, additional demands on the system (and subsequently the vendor) because of custom modifications and interfaces have increased the complexity of the system. Our system is no longer “just like all the other systems” the vendor has installed and is supporting. Tighter control and better tracking of the vendors actions are necessary to ensure that what one support person does will not interfere or conflict with what another support person has done. We can no longer assume that the vendor support personnel, through their own revision management system, are able to deal with the unique environment which has been created in our institution and ensure the continued availability and integrity of the system without our direct involvement.

During

Responsibility for developing and implementing a vendor remote access procedure in our organization fell directly to me functioning in the role of Information Technology Officer. While no formal team existed for the project, input and assistance was solicited from our Network Administrator, our Operations Manager, and our Internal Auditor to ensure that as many points of concern as possible could be considered.

The procedure to be developed was defined by the following statement:

> A vendor remote access management system to provide
>
> 1) an auditable, internally controlled method of granting access to the appropriate vendor support staff members who were deemed to require such access
>
> 2) by authorized individuals within our organization,
>
> 3) to restrict that access to the specific systems required during a specific time frame,
>
> 4) to grant only the level of authority necessary on the network and systems, and
>
> 5) to enable our internally authorized people to affect the granting of access without the intervention of the internal information technology support staff or network administrators.

Our initial step was to review our policy concerning remote access in general and determine what vendor remote access procedures needed to be changed. It was determined that the policy needed to be modified by differentiating vendor remote access from employee remote access. Therefore a specific section was added to specify the five points above.

We then made an inquiry of our external auditors to get their guidance on specific procedures that they would recommend for our particular need. They were not able to provide a model procedure for us to consider.

We then searched for articles and procedures on the Internet that could be used as a starting point for developing the mechanics of our desired system. While many general level articles and documents were located, none were able to provide us with a specific start point. Nonetheless, the documents provided us with a general base of recommendations and best practices:

Eliminate uniform default passwords, disable modems when not required, use a remote-access-server (RAS) that allows for the use of strong identification and authentication tools, and partition or segmenting of the physical network to restrict access (Kabay).

Use the Remote Access Control Policy in Windows 2000 to set dial-in privileges, no unauthorized RAS or other access point are allowed on the network, set appropriate auditing polices withing the Group Policy of the Windows 2000 Active Directory, and make all internal machines subject to scans by internal security tools (Dodds and Pfeil).

Manage login accounts to the network from a return on investment (ROI) perspective and recognize the costs associated with properly managing accounts with remote access privileges (Armstrong).

Know who is accessing your network, use authorization to manage fine-grain access to resources, and establish Groups to associate roles with different type of users (Mackey).

The key to secure remote access is a combination of identifying threats and responding appropriately, match data sensitivity with adequate countermeasures, perform risk assessment, and layer on protection (Stephenson).

We then made an inquiry with our vendor as to how other financial institutions managed vendor remote access to their systems. Surprisingly, no other customer of our vendor had developed a procedure beyond what the vendor had put in place on our system. The vendors support coordinator was however very open and receptive to working with us to develop a procedure. Their feeling was that without us putting such a procedure together that we were entrusting them with a more responsible security role for our network and systems than they were prepared to assume.

Feeling a bit like Columbus venturing off to discover the New World, we put on our pioneer hats and headed West with a dream for a better life and a more secure network.

Establishing Who, What, When, Where, Why, and How

Identifying the specific individuals within our organization (user) who would be given the authority to initiate support requests to the vendor and subsequently grant the vendor access to the systems was our first task. Through conversations with our Deposit Operations Manager, the users identified were those who had already been made responsible for specific operational areas of

the system, had been provided with elevated levels of operational access to the system, and who had been trained by the vendor on the daily operations of the system. This was a very limited group of individuals that exhibited a good understanding of the need for increased accountability of the vendor on our system.

Identifying the specific vendor support personnel (vendor) who would need access to the systems presented a potentially high maintenance function of creating and maintaining a list of all support personnel authorized by the system provider to support our systems. While the ideal solution would be to create specific logins accounts for each member of the vendor support team, the reality of creating 30+ specific accounts to match to each individual vendor who would have to be identified and managed was deemed to be too much of a potential failure point in the overall vendor access management system.[5] A compromise was reached which actually increased the auditable nature of the overall system: a naming convention for a set of vendor support accounts where a concatenation of the users name and an enumerator would yield a vendor login account such as “johnd-sup01”, “johnd-sup02”, and “johnd-sup03” (xxxxxx-supnn) for the user John Doe. A set of three vendor support accounts was established in the Active Directory for each user.

In order to ensure that only users authorized to grant vendor access were enabled to act on the vendor accounts, a global group named “Support Account Operators” was created in the Active Directory controls of the domain and all of the “xxxxxx” user accounts associated with the vendor accounts were made members. This group was used to restrict network resource access to a shared folder containing batch and log files on a server common to the users.

In order to facilitate the granting of specific rights on specific servers in a simple and consistant manner, a global group named “Vendor Support” was established in the Active Directory and all of the “xxxxxx-supnn” user accounts were made members. This allowed us to make a single change on a per server basis: adding the “Vendor Support” group to the “Administrators” group in the local users and groups controls on only the servers they were to be granted access and authority to. The systems that the vendor needs access to run on Windows 2000 Server™ based servers acting as member servers in our Active Directory environment. Access and authority can be controlled by way of locally defined users and groups as well as by way of domain level user accounts and groups. Because of the level of integration of the systems integration with the Windows environment, it was determined that the vendor would need administrative level privileges on the local servers they would be granted access to but not on the network as a whole.

By adding the domain global group “Vendor Support” to the local “Administrators” group on the specific servers listed in the “Log On To…” option of the vendor account properties the vendor was provided full rights to perform their functions

while preserving tight control of the domain level Administrator access and privilege.

Because the Active Directory properties for the vendor account objects that we created were too generally maintainable and open, the “xxxxxx-supnn” vendor account properties were each modified as follows:

Property Action	Purpose
Entered a Description for the account	To increase the identification of the vendor account object during audit reviews of the Active Directory account objects
Set the account to “Disabled”	The default desired status of the account to enable a “disabled objects” report
Set the “User Cannot Change Password” option	To prevent the account from being modified by the vendor
Set the “Log On To…” option to include the NETBIOS names of only the specific servers the vendor personnel would need to access	To restrict the physical machines the account could be used to access
Set the “Logon Hours…” option to Logon Denied for all hours of all days	The default desired time of day that the account could be used
Set the “Account Expires” option for a date ending sometime in the past	The default desired date after which the account cannot be used
Set the “Dial-in” value for Remote Access Permission to “Allow Access”	To enable the account to be used as a dial-in remote account
Set the “Security” values by:	
a) Deselecting “Allow Inheritable Permissions from Parent to Propagate to this Object” and copying the previously inherited permissions to this object	To stop the implying of rights from the parent object onto the account and allow for specific additions and deletions of rights
b) Removing the Account Operators group	To prevent “non Support Account Operators” from managing the vendor account
c) Removing the Authenticated Users group	To prevent the account from being modified by the vendor or anyone else
d) Removing the Everyone group	To prevent the account from being modified by the vendor or anyone else
e) Adding the associated user account with Full Control permission	To enable the user that “owns” the vendor account to manage the account

Providing a means for our users to actually act on the accounts they owned was required. While we could have enabled our users to directly access one of the Active Directory controllers on the network to run the "Users and Computers" management console, that would have created a whole set of additional problems that would have had to have been addressed. In order to minimize the training of the users, to simplify and standardize the actions the users would need to perform to act on the accounts they owned, and to minimize the overall number of people who would need to access the domain controls of the Active Directory domain controllers, a command level batch file (attached) was written to act as an interface between the users and the Active Directory.

Through a series of prompts and responses while executing the command batch file, a parameterized "NET" command is built, echoed into a log file along with the date, time, vendor name, and reason and then executed to manipulate the vendor account object properties in the Active Directory. The parameters gathered are:

- the enumerator for which vendor account to activate,
- the expiration date for the vendor account,
- the start time for the time of day restriction,
- the end time for the time of day restriction,
- the specific name of the vendor support person authorized,
- a brief comment to reflect the reason the vendor required access, and
- the password to be used with the vendor account

The command batch file and the required support programs and files were then placed in a shared folder on a server and the users were shown how to access the folder and run the interactive process. The security property of the shared folder was modified to restrict access to the folder and its contents to only the users who were members of the "Support Account Operators" group.

These changes effectively:

1) "joined" the vendor account to the user account in such a way as to ensure that the only time a vendor account could be used was through the specific action of the user who owned the account,
2) specified a set of controlling factors to place the object under tight control, and
3) logged specific information about the circumstances surrounding the activation of the vendor account

The establishment of a regimented, controlled system for restricting access to secured systems will only be effective if all avenues for gaining remote access

fall within the controls of the system. Uncontrolled access points will negate the system. To further enhance the ability to audit the remote access system, the avenues of access should be reduced to only the required points.

A Cisco remote access server equipped with multiple modems and lines was established to replace the multiple single modem access points that existed. The CiscoSecure ACS TACACS+ was deployed on the network and configured to use the Active Directory as an external database. This ensured that the vendor accounts would only authenticate according to the status, password, time of day, account expiration, and dial-in properties set for them in the Active Directory.

By configuring the ACS to log both passed and failed authentication attempts and to create accounting logs that reflect the start and stop times of access and volume of usage during the sessions, a clear picture of how and when access to our network was achieved is preserved. Entries in the ACS log can be directly correlated to the log file entries created during the user process of activating an account.

Communications with our vendor was maintained throughout the process of setting up our procedure. This allowed them to test the process rigorously prior to the final implementation. When the implementation took place, the vendor support personnel were not only aware of how the system worked but were in support of our efforts to control their access to the system.

After

Implementation of the defined procedure met our goals for controlling vendor support access to our systems. It placed us in direct control of our systems and absolved the vendor from most of the security burden which they had been responsible for in the prior arrangement.

Who is accessing our system can now be determined by reviewing our remote access and account activation logs. The transitive nature of the vendor account provides some assurance that the account is being used only by the person intended.

Who inside our organization authorized the vendor access can now be determined by reviewing our account activation logs. The means by which the user is identified through the vendor account activation process precludes unauthorized users from being able to activate a login account for the vendor to use.

What the vendor was supposed to be doing while connected to our systems can be generally determined by reviewing our account activation logs. Specifically what the vendor is doing is now restrictively controlled and can be audited in the system event logs. This information, combined with information the vendor records through a logging function internal to the software being supported, provides a clear purpose for the system access.

When the vendor accessed the network and the systems and for how long can now be determined by reviewing our remote access logs for network access times and through entries logged in the system event logs on the servers accessed.

Where on our network the vendor was working is now not only restrictively controlled but logged in the system event logs. Where the vendor originated the access from is recorded in the security event log on the servers they access.

Why the vendor needed access to our systems can be determined by reviewing our account activation logs for notes made by the user that authorized the access.

How the vendor accessed the network can now be readily determined by virtue of the fact that only one point of access exists and is restrictively controlled.

Additional areas to be considered in enhancing the controls include:

1. Implementing a VPN solution between the vendor network and our network would eliminate the need and risk of a modem pool and increase integrity by having only encrypted transmissions.

2. Increase security awareness training of our internal users who authorize vendor access so that vendor access is viewed as the risk that it is and can be limited as much as possible.

References

1. Federal Financial Institutions Examination Council. IT Examination Handbook Information Security Booklet. December 2002.
2. IT Governance Institute, Information Systems Audit and Control Association. COBIT: Control Objectives for Information and related Technology. 2000.
3. Kabay, M. E.. “Controlling Vendor Access to Production Systems.” Network World Security Newsletter. 06/05/00. URL: http://www.nwfusion.com/newsletters/sec/2000/0605sec1.html (May 10, 2003).
4. Dodds, Tom and Pfeil, Ken. “Security Considerations for End Systems.” Best Practices for Enterprise Security. URL: http://www.microsoft.com/technet/security/bestprac/bpent/bpentsec.asp (May 10, 2003).
5. Armstrong, Illena. “Access Management Part One: Sound ROI With Security Benefits.”October 2002. URL: http://www.scmagazine.com/scmagazine/2002_10/cover/index.html (May 10, 2003).
6. Mackey, Richard. “Access Management Part Two: Authorizing Your Users.” October 2002. URL: http://www.scmagazine.com/scmagazine/2002_10/cover/index2.html (May 10, 2003).
7. Stephenson, Peter. “Securing Remote Access.” URL: http://www.networkcomputing.com/602/602work2.html (May 10, 2003)

Interactive Batch File for User Activation of Vendor Account

```
****************************** Start of File ****************************************

@echo off
cls

REM   This section pulls the username vaule from the environment and presents
REM    a “Menu” for the user to choose from
:getacct
echo 1)  %username%-sup01
echo 2)  %username%-sup02
echo 3)  %username%-sup03
echo.
goto getvars


REM   This section progresses through a series of prompts and inputs
REM    accumulating values in the environment
:getvars
.\input     "Which support account (Enter 1, 2 or 3) ?" acct
call .\input.bat
if X%acct%==X goto getacct

echo.
echo The account should be set to expire on the next day.
echo  Such as - today is %date%.  Set the expiration for tomorrow.
.\input "When should the account expire? (use slashes: %date%) " exp
call .\input.bat

echo.
.\input "What time should access start (use 2p for 2 in the afternoon): " timstrt
call .\input.bat

echo.
.\input "What time should access end (use 5p for 5 in the afternoon): " timend
call .\input.bat

echo.
.\input "Enter the password to be used for this account " pwd
call .\input.bat

echo.
.\input "Who is this support account being activated for? " who
```

```
call .\input.bat

echo.
.\input "Why is this support account being activated? " why
call .\input.bat

del .\input.bat
goto sendline

REM    This section sends to values which have been gathered to a log file
REM     and then executes the “NET” command substituting the values into the
REM     commands as parameters
:sendline
date/t >> .\SUP-Logs.txt
time/t >> .\SUP-Logs.txt
echo Activated for - %who% >> .\SUP-Logs.txt
echo Reason - %why% >> .\SUP-Logs.txt

REM…. This is a single line command that has been wrapped
echo user %USERNAME%-sup0%acct% %pwd% /active:yes /expire:%exp%
/times:m-su,%timstrt%-%timend% /domain >> .\SUP-Logs.txt

echo ..... >> .\SUP-Logs.txt

REM…. This is a single line command that has been wrapped
net user %USERNAME%-sup0%acct% %pwd% /active:yes /expire:%exp%
/times:m-su,%timstrt%-%timend% /domain

goto end

:end

****************************** End of File ****************************************
```

Notes:

1) Some of the variables and commands used in this batch require a Windows 2000 environment.
2) The INPUT executable program is a third party product that takes a text parameter to use as a prompt and a variable name to receive a value, receives the input from the user, and creates a batch file named INPUT.BAT that when executed sets the environment variable with the value provided by the user.