Global Information Assurance Certification Paper

Baselining a CVS Server

GIAC System and Network Auditor (GSNA)
Practical Assignment Version 4.0 Option 1 Topic 2
"Baselining"

Hemant Gautam
March 29, 2005

Abstract

This paper will develop a baseline to audit Concurrent Versions System (CVS).

Part 1 of this paper will identify what the baseline should address and then will summarize the baseline elements specific to CVS.

Part 2 will develop the baseline. Each baseline item identified in part 1 will be detailed.

Finally, Part 3 will contain the procedures for testing against each baseline item.

Table of Contents

Abstract

Task 1 Identify the Baseline
1.1 Introduction to CVS
1.2 Scope
1.3 Identify the System
1.4 Components of a Baseline
1.5 Importance of each element of Baseline to system's security posture

Task 2 Develop the Baseline

Task 3 Procedure for Testing against the Baseline

List of References

Task 1 Identify the Baseline

1.1 Introduction to CVS

CVS or Concurrent Versions System is an Open Source (released under GNU GPL[1]) version control system that "allows multiple users to access, modify and update simultaneously a set of files in a directory or several directories, without getting in each other's way, most of the time."[2] Generally, CVS is used in software development environment where there are many projects simultaneously going on and each developer contributes to one or more projects.

CVS is based on Client-Server model. The server stores all the files for which version control is required in a central place called "repository". The clients connect to this repository and they "check-out" relevant file on which they are working and after doing modifications they "check-in" the file back in the repository. Client can be Windows or UNIX based.

Clients can connect to CVS via three methods:

1. pserver (Password Authentication Server)
2. Tunneling pserver inside SSH
3. SSH

1.2 Scope

As the title of this paper suggests, this will develop a baseline for testing a CVS server. CVS is an application which is dependent upon many sub-components like Operating System on which it will be installed, Communication Protocols it will use for client-server interaction, Networking Infrastructure which will help in client-server communication, Networking Environment (hostile or friendly), Client's workstations. For each of these sub-components there can be a baseline but in developing that baseline the focus of this paper would be lost. Therefore, this paper will make the assumption that all these subcomponents have been configured to match the best practices followed for hardening or as per organization's security policy.

1.3 Identify the System

This paper assumes the existence of a fictitious organization called KwikSoft involved in Software Development. Currently they are using CVS for 3 different software development projects and team size for each project is 12.

1. https://ccvs.cvshome.org/. For GPL http://www.gnu.org/licenses/gpl.html

2. Dick Grune, http://www.cs.vu.nl/~dick/CVS.html#History

For all the three projects they are using CVS Ver 1.11.18 running on Red Hat Enterprise Linux 3.0. The developer's workstations are Windows 2000 Professional and they use TortoiseCVS[3] which connects to repository via pserver.

1.4 Components of a Baseline

Base lining Process

Security Definitions

Information Security is optimally defined using three keywords - Confidentiality, Integrity and Availability (CIA):

Confidentiality – To safeguard against unauthorized access;
Integrity – To safeguard against modification;
Availability – To safeguard against unavailability;

In the project related to CVS, C,I, and A plays a significant direction towards baselining definition.

Base lining is a step-by-step process with the following objectives:

a. Identify subcomponents of a target environment (application, OS, Database) to identify weaknesses;
b. Identify those components which results in compromise of C, I, A or a combination thereof.
c. Identifying the subcomponents and associate a weightage in order to indicate the significance of the subcomponent in the overall security posture.

3. www.tortoisecvs.org

1.5 Importance of each element of Baseline to system's security posture

Sr. No.	Elements	Impact on
1	Patch management for CVS	C,I,A
2	Only authorized users have write access to CVSROOT	C,I
3	Edit the CVS passwd file in place. It should not be in checkout list file.	C,I
4	Pserver should run as non-root account	I,A
5	Connecting to Repository securely over the Network	C
6	Client's passwords should be stored securely	C
7	Fallback behavior in pserver should be disabled.	C
8	If CVS users are also System users then their passwords should be different.	C
9	Comments on CVS config files while checking in	I
10	Userid should be different for each developer	I
11	To edit the configuration files in CVSROOT checkout a local copy and then modify	I
12	Restrict the access to pserver.	I
13	Restrict port 2401 at the Network Perimeter	I
14	Regular Backups of CVS Repository	A

Task 2 Develop the Baseline

The following resources were used to develop the baseline:

a. CVS version 1.11.18 running on Red Hat Enterprise Linux 3.0
b. TortoiseCVS on Windows 2000 Professional.

Each element of baseline has been assigned a weightage which determines its importance to overall baseline. A four point scale has been used viz. 0.25, 0.5, 0.75 and1 which range from low importance (0.25) to high importance (1). For example a weightage of 1 to a specific Baseline element demonstrates a higher degree of compliance required for CVS protection, compared to lower figures.

Note for Auditors

While evaluating CVS protection using these baselines, the Auditors should take note of the *Importance to overall Baseline* grading (.25, .5, .75 and 1.0) in order to understand the scale of compliance necessary for protecting the CVS. Auditors should also raise serious concerns or refer to the system as ineffective in case the *Expected Value does not* match with audit findings.

Baseline elements

Baseline element # 1	
Description	Patch management for CVS A patch can be a security bug fix or some kind of performance enhancement. It is absolutely necessary for an organization to stay up-to-date with the patches released by the vendor.
References	NIST "Security Self-Assessment Guide for Information Technology Systems" Appendix A System Questionnaire item 10.3.2, Page A-33 The Twenty Most Critical Internet Security Vulnerabilities, http://www.sans.org/top20/#u4

Importance to overall Baseline	1
How to evaluate	Check the Change Management document for the last upgrade i.e. when the CVS Server was upgrade to version 1.11.18
Expected value	Patches, especially security updates released by the vendor should be installed as soon as possible.

Baseline element # 2

Description	**CVS user doesn't have write access to CVSROOT** "..you must control the permissions on this directory as tightly as the permissions on '/etc'." Configuration files in CVSROOT directory are the core of CVS, only authorized users should have permission to modify these files. Ideally there should be one group who has the ownership of CVSROOT and included in this group are only trusted users who will have the task of administering CVS repository.
References	Version Management with CVS for CVS 1.11.18 per Cederqvist et al Page 25, Section 2.9.3.3 O'reilly – Essential CVS, Chapter 6-Section 6.5 Open Source Development with CVS, Chapter-3, Page 114
Importance to overall Baseline	1
How to evaluate	Step 1: Find out the user id of persons who are authorized to edit the configuration of CVS server. Step 2: In the CVS server check the permission on the $CVSROOT directory.
Expected value	Only authorized users should have write access to CVSROOT directory.

Baseline element # 3

Description	Edit the CVS passwd file in place. It should not be in checkout list file. If fallback behavior is disabled then pserver uses the $CVSROOT/CVSROOT/passwd file to authenticate users. This file has user name, passwords and corresponding system user names. Therefore this file should be edited in place and should not be checked-out like the other configurations files in CVSROOT directory. If you want to maintain a customized administrative file with CVS (like other configuration files) then you have to put its filename in one of the CVS configuration file 'checkoutlist'.
References	Version Management with CVS for CVS 1.11.18 per Cederqvist et al Page 146, Section C.7.
Importance to overall Baseline	.75
How to evaluate	Check the 'checkoutlist' file in $CVSROOT/CVSROOT directory, whether it contains the word 'passwd'.
Expected value	The 'passwd' file should not be in 'checkoutlist'.

Baseline element # 4

Description	Pserver should run as non-root account When you are configuring pserver in xinetd.conf you have to specify a system username which will be used to run the pserver. Its not a good practice to run pserver with root id. In case there is a buffer overflow in pserver and if it is exploited then it will run with the user id of root! This will compromise the whole system.
References	O'reilly – Essential CVS, Chapter 8-Section 8.8
Importance to overall Baseline	.75
How to evaluate	Check the xinetd.conf file for user id.
Expected value	The username for pserver should be a non-root.

Baseline element # 5

Description	Connecting to Repository securely over the Network Client should authenticate to the repository in such a manner that their password is not sent in clear text. If it is sent in clear text then it might be subject to sniffing attack.
References	NIST Publication "Generally Accepted Principles and Practices for Securing Information Technology Systems" Page 44 "Secure Transmission of Authentication Data"
Importance to overall Baseline	.75
How to evaluate	Check the CVS documentation for encryption method used by 'pserver'.
Expected value	Password sent over the network should be suitably encrypted to thwart any sniffing attack.

Baseline element # 6

Description	Client's passwords should be stored securely. The password used by the user to authenticate with the CVS server should be stored in a secure manner to prevent unauthorized access.
References	NIST Publication "Generally Accepted Principles and Practices for Securing Information Technology Systems" Page 44 "Restrict Access to Authentication Data"
Importance to overall Baseline	.75
How to evaluate	Check the stored password for location and encryption.
Expected value	Password should be stored in a secure manner.

Baseline element # 7

Description	If using pserver then disable fallback behavior When a client connects to the CVS repository using pserver he can be authenticated either using the username and password in '$CVSROOT/CVSROOT/passwd' or using the system username and password as contained in '/etc/passwd'. First the CVS will check the '$CVSROOT/CVSROOT/passwd' file and if username is not found then the '/etc/passwd'. This is falling back to system authentication. This fallback is a security risk because in the case of authentication using pserver the password travels across the network in plaintext.
References	Version Management with CVS for CVS 1.11.18 per Cederqvist et al Page 23.
Importance to overall Baseline	.75
How to evaluate	Check the value of 'SystemAuth' in the 'config' file.
Expected value	The fallback behavior should be disabled.

Baseline element # 8

Description	If CVS users are also System users then their passwords should be different. In some cases an organization might use CVS server for other functions also, like for ftp, development web server etc. In that case a normal CVS user might also be a system user i.e. logging directly into the system, if it is so then the password used in both case should be separate because passwords travel in plaintext when user connects to CVS via pserver.
References	O'reilly – Essential CVS, Chapter 8-Section 8.7.4 Open Source Development with CVS, Chapter-3, Page 99
Importance to overall Baseline	.75
How to evaluate	Ask the person who is responsible for administration of CVS server.
Expected value	If a CVS user also logs in to the system for other purposes then the passwords should be different.

Baseline element # 9

Description	Comments on CVS configuration files while checking in Accountability is a part of integrity. It helps in establishing audit trails. Whenever a CVS administrator checks-in after modifying a file it is imperative that he should sufficiently describe in his comment why he modified/added a file. In case in the future CVS breaks reporting some error in configuration files then it's easy to trace back the file modifications done by CVS administrator.
References	NIST Publication "Generally Accepted Principles and Practices for Securing Information Technology Systems" Page 50, Section 3-13
Importance to overall Baseline	.75
How to evaluate	Check the File history with the help of TortoiseCVS.
Expected value	Every modification done to CVS configuration files should be supported by sufficiently detailed comments.

Baseline element # 10

Description	Userid should be different for each developer This is another element which contributes to accountability and audit trails. Without separate id for each developer it's impossible to establish who did what. With different Userid it will be easier to check for attempt of unauthorized access and any malicious intent thereof.
References	NIST Publication "Generally Accepted Principles and Practices for Securing Information Technology Systems" Page 50, Section 3-13 "Individual Accountability"
Importance to overall Baseline	.75
How to evaluate	Ask the person who is responsible for administration of CVS server.
Expected value	Each CVS user must have his own ID.

Baseline element #11

Description	To edit the configuration files in CVSROOT checkout a local copy and then modify Files in the CVSROOT directory are CVS configuration files which can modify the way CVS server functions. It is therefore necessary that they should be modified in the same way as other files in the repository i.e. checkout-modify-checkin. This will ensure that if a configuration file in CVSROOT modified incorrectly then it's easy to check in working old version.
References	Version Management with CVS for CVS 1.11.18 per Cederqvist et al Page 17, Section 2.4.1.
Importance to overall Baseline	.5
How to evaluate	Use 'cvs history' command to verify this element.
Expected value	Configuration files in CVSROOT should be modified using the usual checkout-modify-checkin method.

Baseline element # 12

Description	Restrict the access to pserver. Only those workstations which are used by developers should be allowed to access pserver. Since pserver uses xinetd and xinetd already has the feature to restrict the access based on host names or ip address, this restriction can be achieved easily.
References	Principle of Least Privileges, http://hissa.nist.gov/rbac/paper/node5.html
Importance to overall Baseline	.5
How to evaluate	Check the xinetd.conf file for restrictions.
Expected value	Only IPs of those users who are authorized to access the CVS server.

Baseline element # 13

Description	Restrict port 2401 at the Network Perimeter To restrict anybody from exploiting a potential vulnerability in CVS server it is recommended that Port 2401 should be blocked at the perimeter. It can either be blocked at the router or at the firewall or both.
References	The Twenty Most Critical Internet Security Vulnerabilities, http://www.sans.org/top20/#u4
Importance to overall Baseline	.75
How to evaluate	Check the Perimeter device's configuration.
Expected value	Port 2401 should be blocked at Network Perimeter.

Baseline element # 14

Description	Regular Backup of CVS Repository Backup is an insurance against: a) Hardware failure b) Disaster (Fire, Floods) c) Application failure d) User error Frequency of backup depends upon the nature of data how static or dynamic data is and also upon recovery requirements. Dynamic data requires frequent backups as opposed to static data which doesn't need frequent backups. Data in a CVS repository is very dynamic constantly accessed and changed by the users.
References	Implementing Backup and Recovery The Readiness Guide for the Enterprise- Introduction and Chapter 1 "Why is the Data Backed Up, Chapter 2 Frequency of Backups.
Importance to overall Baseline	.75
Steps taken to evaluate	Step 1: Check the Data Backup Policy. Step 2: Check the Log created by 'cron'.
Expected value	Data should be backed up daily. Full Backup on weekends and incremental on weekdays.

Task 3 - Procedure for Testing against the Baseline

Baseline element #1
Description Patch Management for CVS
Whether requires 'root' privileges to check? No

Testing Procedures. 1. Log in to the CVS server. 2. Execute the 'cvs -v' command from the console. This command will display the version of CVS currently running. Note down the version. 3. Go to www.cvshome.org and check if there is any security update in the newer version. 4. If there is any security update then check with the CVS administrator for plan to update to the current version. 5. If there is unreasonable time gap between the newer version and date of audit, then it means that they are not serious about the security updates.
Indications of Non-Compliance Time gap implies No Patch Management Policy

Baseline element #2
Description Only authorized users have write access to CVSROOT
Whether requires 'root' privileges to check? No
Testing Procedures. Procedure 1 1. Login to the CVS server. 2. Execute the 'ls -l' command to check the permissions for the $CVSROOT/CVSROOT directory. Note down the permissions as well as username and group name who owns the directory. 4. Find out the members of the group by executing the following command 'cat /etc/group | grep 'group_name' 5. Find out from the CVS administrator which user account he uses for CVS administration. 6. This can be cross verified by checking with the group member(s) noted in step 4 and user as in step 2. Procedure 2 This baseline element can also be verified from the output of baseline element # 11 using the following procedures: 1. Check if any configuration file was modified remotely (Step 3 in Baseline Element # 11), note down the name of the file. 2. Go to workstation used by CVS administrator and checkout a copy of $CVSROOT/CVSROOT with the same Userid used by him. 3. Go to CVSROOT directory in the workstation and right click on the same file noted in step 1, then click on CVS and then History. The popup box will show the name of the user who modified the file together with any comments. 4. Now this user name can be cross verified in the same way as in step 6 above.

Indications of Non-Compliance
If the user id actually used to modify the configuration files does not matches with the user id found out in Step 2 & 4 in Procedure 1 then it implies that permissions on $CVSROOT/CVSROOT directory are not tightly controlled.

Baseline element #3
Description Edit the CVS 'passwd' file in place. It should not be in checkout list file.
Whether requires 'root' privileges to check? Yes
Testing Procedures. 1. Login to the CVS Server. This procedure will require either the root login or login of the user who is authorized to modify/read the CVS configuration files. 2. Execute the following command from $CVSROOT/CVSROOT/ directory 'cat checkoutlist' This command will print the content of 'checkoutlist' file. The 'passwd' file should not be in the contents. 3. This element can also be verified by executing the following command : 'cvs history -a -e | grep passwd' 'history' can be used to track use of checkout, commit, rtag, update and release commands. '-a' will show data for all users '-e' will show all record types. If the output of this command has keyword '<remote>' then it means that this file was modified remotely.
Indications of Non-Compliance 'passwd' file modified remotely implies Username/password of all users can be compromised.

Baseline element #4
Description Pserver should run as non-root account
Whether requires 'root' privileges to check? No

Testing Procedures.
1. Ask the CVS administrator whether they have made the entry in /etc/xinetd.conf file or have created separate file in /etc/xinetd.d/ directory.
2. Login to the CVS server.
3. Case 1 - Execute the following command : cat /etc/xinetd.conf and look for entry made for pserver.
   Case 2 - Execute the following command : cat /etc/xinetd.d/name_of_file
Entries in xinetd.conf or as a separate file in /etc/xinetd.d/name_of_file are of same format like below:

```
service <service_name>
{
        statement 1 = item 1 item 2 ....
        statement 2 = item 1 item 2 ....
        .......
}
```

The line we are looking for will be something like

```
service pserver
{
                .....
                user = <user_name>
                .....
}
```

Indications of Non-Compliance
Running with Root privileges implies a buffer overflow in pserver can compromise the whole system.

Baseline element #5
Description Connecting to Repository securely over the network.
Whether requires 'root' privileges to check? No
Testing Procedures. 1. Check the CVS documentation for the encryption method used by CVS to encrypt the user's password when they travel across the network.
Indications of Non-Compliance and effects. Weak Encryption algorithm implies its easy to decipher the password

Baseline element #6
Description Client's passwords should be stored securely.

Whether requires 'root' privileges to check? No
Testing Procedures. 1. Check the CVS and TortoiseCVS documentation for the encryption method used for storing the client's password on the workstation from which he logs in to the CVS server.
Indications of Non-Compliance and effects. Weak Encryption algorithm implies it's easy to decipher the password

Baseline element #7
Description Fallback behavior in pserver should be disabled.
Whether requires 'root' privileges to check? Yes
Testing Procedures. 1. Login to the CVS server. This procedure will require either the root login or login of the user who is authorized to modify CVS configuration files. 2. Execute the following command from the $CVSROOT/CVSROOT/ directory: 'cat config | grep SystemAuth' 3. If the output is: '#SystemAuth=no' then it means that fallback behavior is enabled. Ideally this line should not be commented. '#' sign indicates that this is a comment. Removing the '#' sign will disable the fallback behavior.
Indications of Non-Compliance Fallback behavior implies passwords required to log in to the system can be sniffed off the wire.

Baseline element #8
Description If CVS users are also System users then their passwords should be different.
Whether requires 'root' privileges to check? Not Applicable
Testing Procedures. 1. This can only be verified by asking the CVS administrator, as the passwords are stored in encrypted form.
Indications of Non-Compliance Passwords are same implies whole system can be compromised by sniffing the passwords required to log in to the system.

Baseline element #9
Description Comments on CVS configuration files while checking in.

Whether requires 'root' privileges to check? No
Testing Procedures. First verify which CVS configuration file was modified: 1. Login to the CVS server. Any system user can complete this procedure. 2. Execute the following command: 'cvs history -a -e | grep CVSROOT' 'history' can be used to track use of checkout, commit, rtag, update and release commands. '-a' will show data for all users '-e' will show all record types. This command will extract any configuration file modified by any user in $CVSROOT/CVSROOT/ directory. 3. If the above command produces any output then it means that configuration files have been modified. Now to verify whether the user who modified that configuration file entered any comments take the following steps: 1. Go to the CVS Administrator's workstation from which he logs in to CVS server for administrative purposes. 2. Check-out a copy of CVSROOT folder. 3. Locate the modified file in the CVSROOT folder, right click on that file and click CVS then click on History. The dialog box will show the complete history of that file together with comments.
Indications of Non-Compliance and effects. Missing Comments leads to difficulties in troubleshooting if configuration changes breaks down the CVS server.

Baseline element #10
Description Userid should be different for each developer.
Whether requires 'root' privileges to check? Yes
Testing Procedures. 1. Login to the CVS server. 2. The $CVSROOT/CVSROOT/passwd file lists all the users that are authorized to login to the CVS server. 3. This procedure needs cross check from the CVS administrator. It may also happen that an organization uses pserver but does not uses $CVSROOT/CVSROOT/passwd file (fallback authentication enabled-Baseline element #7) but uses locally created system users in that case the element can be verified by asking the CVS administrator whether Userid for each developer is different or not.

Indications of Non-Compliance Common Userid leads to difficulties in tracking who did what and any malicious intent thereof.

Baseline element #11
Description To edit the config files in CVSROOT checkout a local copy and then modify.
Whether requires 'root' privileges to check? No
Testing Procedures. 1. Login to the CVS server. Any system user can complete this procedure. 2. Execute the following command: 'cvs history -a -e | grep CVSROOT' 'history' can be used to track use of checkout, commit, rtag, update and release commands. '-a' will show data for all users '-e' will show all record types. This command will extract any configuration file modified by any user in $CVSROOT/CVSROOT/ directory. 3. If the above command produces any output then it means that configuration files have been modified. Now to check whether they were modified locally or remotely check for '<remote>' keyword in each line of the output. '<remote>' keyword indicates that file was modified remotely.
Indications of Non-Compliance Configuration files modified locally leads to difficulties in tracking who modified which files and why.

Baseline element #12
Description Restrict access to CVS server.
Whether requires 'root' privileges to check? No

Testing Procedures.
1. Ask the CVS administrator whether they have made the entry in /etc/xinetd.conf file or have created separate file in /etc/xinetd.d/ directory.
2. Login to the CVS server. Any normal system user can check this.
3. Case 1 - Execute the following command : cat /etc/xinetd.conf and look for entry made for pserver.
   Case 2 - Execute the following command : cat /etc/xinetd.d/name_of_file
Entries in xinetd.conf or as a separate file in /etc/xinetd.d/name_of_file are of same format like below:

```
service <service_name>
{
        statement 1 = item 1 item 2 ....
        statement 2 = item 1 item 2 ....
        .......
}
```

The line we are looking for will be something like

```
service pserver
{
                .....
                only_from = <host_name or IP address>
                .....
}
```

Indications of Non-Compliance
No access restrictions to pserver implies it can be prone to malicious access attacks.

Baseline element #13
Description Restrict port 2401 at the Network Perimeter.
Whether requires 'root' privileges to check? Not Applicable
Testing Procedures. This can be verified at two different places: 1. At the Router (if used)- For example in case of Cisco Routers look at the access-list. For e.g. typically it will be like this: access-list 102 deny tcp any any eq 2401. 2. At the Firewall- Look for a rule that deny access to port 2401.
Indications of Non-Compliance Open port 2401 implies a vulnerable CVS server can be exploited from outside the organization's network i.e. internet.

Baseline element #14
Description Regular backup of CVS Repository.
Whether requires 'root' privileges to check? Yes.
Testing Procedures. 1. Login to the CVS Server 2. Check the 'crontab' file in /etc/ for how the backup is scheduled. Note down the time of backup from the 'crontab' file. Also, take a note of the filename and full path where it is created. Any normal system user can read 'crontab' file. 3. Check the 'cron' file in /var/log/. Whenever any command or script is executed through 'crond' daemon it will be logged in this file. Compare the time and command in 'crontab' file to the time and command in /var/log/cron file. If you found the corresponding time and command in /var/log/cron then it indicates that the 'cron' daemon run successfully. 4. Compare the time for the backup schedule in the 'crontab' file to the file creation time of the backup file which is created by this schedule. This will indicate that the backup did completed successfully. Any normal system user can read this output
Indications of Non-Compliance and effects. 1. Entry missing in crontab file implies Irregular backup schedule. 2. Entry missing in cron file implies Irregular backups/Problem with 'cron' daemon 3. Time difference in Step 4 implies Irregular backups

List of References

a. David B. Little and David A., Implementing Backup and Recovery The Readiness Guide for the Enterprise, (Indiana: John Wiley & Sons, 2003)

b. NIST Special Publication 800-14 document entitled “Generally Accepted Principles and Practices for Securing Information Technology Systems" http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf

c. NIST Special Publication 800-26 document entitled “Security Self-Assessment Guide for Information Technology Systems” http://csrc.nist.gov/publications/nistpubs/800-26/sp800-26.pdf

d. SANS, The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts Consensus (Version 5.0 October 8, 2004) http://www.sans.org/top20/

e. Version Management with CVS (2004) https://ccvs.cvshome.org/files/documents/19/532/cederqvist-1.11.18.pdf

f. Jennifer Vesperman, Essential CVS, (California: O'Reilly & Associates,2003)

g. Moshe Bar, Karl Fogel, Open Source Development with CVS (Arizona: Paraglyph Press,2003)

h. David Ferraiolo, Richard Kuhn, Role-Based Access Controls (1995), National Institute of Standards and Technology
http://hissa.nist.gov/rbac/paper/node5.html

i. GNU General Public License
http://www.gnu.org/licenses/gpl.html

j. About CVS License
https://ccvs.cvshome.org/

k. Dick Grune, Concurrent Versions System CVS
http://www.cs.vu.nl/~dick/CVS.html

l. TortoiseCVS Lets you work with files under CVS version control, available under the GPL.
http://www.tortoisecvs.org/

m. CVS first released.
Dick Grune, v06i040: CVS, an RCS front-end (cvs), Part1/2 (1986)
http://groups-beta.google.com/group/mod.sources/msg/2ebab72ac0744fb8?:mod.sources.*=&hl=en&lr=lang_en&ie=UTF-8&c2coff=1&safe=off&rnum=2

n. Encryption method used by CVS and TortoiseCVS for transmitting and storing password.

You can find the source code of encryption method used by pserver to encrypt the password in scramble.c which is in 'src' directory. TortoiseCVS also uses same encryption method when it sends the user's password to the CVS server over the network and when it stores the password on the workstation.
In scramble.c file there is a function called scramble which takes username as input and returns the encrypted password. The first character of the returned encrypted password denotes the scrambling method used. In scramble.c there is only one scrambling method which is denoted by capital letter 'A', but from the comments in the source file it seems that there can be more scrambling methods in the future. The scrambling method uses ASCII value of each character in the password as an index to the character array "shifts[ ]".
For e.g. if the user's password is 'pass' the ASCII value of 'p' is 112 and 112 is used as index to the character array shifts[ ] and the value for index 112 in the array is 48, now we have to check the character corresponding to ASCII value of 58 which next to 48, therefore 'p' becomes ':' in encrypted password. Likewise 'a' ASCII value is 97 and using 97 as index to the character array shifts[ ], its value is 36 and now we have to check the character corresponding to ASCII value of 121 which next to 36, therefore 'a' becomes 'y'. Lastly for 's' ASCII value is 115 and using 115 as index to the character array shifts[ ] its value is 32 and now we have to check the character corresponding to ASCII value of 90 which

next to 32, therefore ‘s’ becomes ‘Z’. So, the ‘pass’ in encrypted form becomes ‘:yZZ‘.
TortoiseCVS store the user’s password in the following registry key (Windows 2000 Professional):

[HKEY_CURRENT_USER\Software\Cvsnt\cvspass]

using same encryption method as described above.