

Sichereres CVS

Carsten Grohmann

11. Juni 2003

Zusammenfassung

Dieses Dokument beschreibt in Grundzügen den sicheren Aufbau eines CVS-Servers unter SELinux. Der Zugang zum CVS kann dabei über `cvs pserver` oder via SSH und `cvs server` erfolgen. Einige Anregungen dazu stammen aus "Chrooted SSH CVS server HOW-TO" [2] von IDEALX [3]. Die Neuerung beruht auf dem Einsatz von SELinux [4] anstelle von chroot, da es ein Mehr an Sicherheit verspricht.

Inhaltsverzeichnis

1	Voraussetzungen und Vereinbarungen	3
2	Verwendete CVS-Gruppen	3
2.1	Projektübergreifende Gruppen	3
2.2	Projektspezifische Gruppen	3
3	Datei- und Verzeichnisberechtigungen	3
3.1	Linux-Berechtigungen	4
3.2	SELinux-Berechtigungen	5
4	Verzeichnisse	5
4.1	CVSDIR/project_p1/cvslock	5
4.2	CVSDIR/tmp	5
4.3	CVSDIR/bin	5
5	Konfiguration	5
5.1	SELinux	5
5.2	CVS	5
5.3	CVSROOT/config	6
5.4	CVSROOT/loginfo	6
5.5	SSH	6
5.6	CHROOT	6
5.7	CVS-Wrapper	6
6	Mögliche Verbesserungen	7

1 Voraussetzungen und Vereinbarungen

Dieses Dokument steht unter der Freien Lizenz für Texte und Textdatenbanken [15] des Instituts für Rechtsfragen Institut für Rechtsfragen der Freien und Open Source Software (ifrOSS) . Zusätzlich ist es untersagt diese Dokument unter jede Art von DRM zu stellen.

Ein lauffähiges System mit CVS und SELinux erleichtert das Verständnis für diese kurz gehaltene Dokumentation.

Das fiktive zugrunde liegende Projekt heißt “p1” und hat die Module “CVSROOT”, “modul1” und “modul2”. Die Gruppennamen lehnen sich an diese Bezeichnungen an, somit ist eine einfache Zuordnung zwischen Projekt, Modul und Gruppe gewährleistet.

2 Verwendete CVS-Gruppen

Um die Zugriffe auf das CVS zu regeln, kam die nachfolgende Gruppenstruktur zum Einsatz. Diese Struktur basiert auf normalen Linuxberechtigungen. Aufgrund der Komplexität der benötigten Berechtigungen ergibt sich die volle Sicherheit vor unberechtigten Zugriffen für CVSROOT erst im Zusammenspiel mit SELinux. Die Berechtigungen aller anderen Module enthält keine Lücken sofern CVS richtig arbeitet, denn der Nurlesezugang wird über CVS geregelt und nicht über die Berechtigungen des Dateisystems.

2.1 Projektübergreifende Gruppen

cvadmin Mitglieder dürfen cvs admin ausführen
cvuser Alle Benutzer mit Zugriff auf das CVS

Um auf das CVS zugreifen zu können, müssen alle Benutzer Mitglied der Gruppe cvsbin sein. Mitgliedern dieser Gruppe ist das Starten des CVS-Server erlaubt. Alternativ kann man bei einem reinen CVS-Server für alle Dateien dieser Gruppe die Berechtigungen so erweitern, daß sie weltles- und ausführbar sind.

2.2 Projektspezifische Gruppen

cvsp1user Alle Benutzer dieses Projekts
cvsp1admin Schreibrechte auf CVSROOT
cvsp1modul1 Schreibrechte auf modul1
cvsp1modul2 Schreibrechte auf modul2
cvsp1ro nur Leseberechtigung

Die Schreibberechtigung für die Benutzer der Gruppe cvsp1ro wird vom CVS verweigert. Dennoch müssen diese Benutzer gleichfalls Mitglieder in der Gruppe mit den Schreibrechten für das Modul, mit welchem sie arbeiten möchten, sein. Die Schreibrechte der Gruppe cvsp1admin auf das Modul CVSROOT, erweitern die Schreibrechte jedes Benutzers zum Beispiel für history um die vollen Berechtigungen für alle Dateien.

3 Date- und Verzeichnisberechtigungen

Für Verzeichnisse wird oft setgid verwendet, damit die Berechtigungen für die Gruppe bei neuen Einträgen erhalten bleiben.

Bei CVSDIR/tmp wird das Sticky-Bit gesetzt, damit nur der Dateieigentümer die Datei löschen darf.

Mehr Informationen zu den Dateisystemberechtigungen befinden sich unter [1].

Mit einer Einschränkung muß man allerdings noch leben. Es ist zur Zeit nicht möglich den security context in Abhängigkeit vom Repository zu wählen. Dieses Manko kann durch die Verwendung unterschiedlicher Gruppen, für die verschiedenen Repositories und Module, ausgeglichen werden.

3.1 Linux-Berechtigungen

Verzeichnist	Benutzer	Gruppe	Berechtigungen	setgid	sticky	rekursiv	Bemerkung
CVSDIR/tmp/	root	cvsuser	770	x	x		Inhalt löschen
CVSDIR/bin/	root	cvsuser	750				
CVSDIR/project_p1/			770	x		x	alle Verzeichnisse
CVSDIR/project_p1/			440			x	alle Dateien
CVSDIR/project_p1/	root	cvsp1user	750	x			dieses Verzeichnis
CVSDIR/project_p1/cvslock/	root	cvsp1user	770	x			Inhalt löschen
CVSDIR/project_p1/repository/	root	cvsp1user	770	x			dieses Verzeichnis
CVSDIR/project_p1/repository/Attic/		cvsp1user				x	sofern es existiert
CVSDIR/project_p1/repository/CVSROOT/	root	cvsp1admin				x	
CVSDIR/project_p1/repository/CVSROOT/			775	x			diese Verzeichnis
CVSDIR/project_p1/repository/CVSROOT/			664				alle Dateien
CVSDIR/project_p1/repository/CVSROOT/history	root	cvsp1user	660				
CVSDIR/project_p1/repository/CVSROOT/val-tags	root	cvsp1user	660				
CVSDIR/project_p1/repository/CVSROOT/.*#*			440				
CVSDIR/project_p1/repository/modul1/		cvsp1modul1				x	
CVSDIR/project_p1/repository/modul2/		cvsp1modul2				x	

Aufgrund der Komplexität der Berechtigungen können diese mit dem Shell-Skript “setcvsrighs” [19] komfortabel gesetzt werden. Die Anpassung an die eigenen Verhältnisse erfolgt nach dem im Tool angegebenen Muster.

3.2 SELinux-Berechtigungen

Dateisystemeintrag	Security Context
/home/cvs(/.*)	system_u:object_r:cvs_dir_t
/home/cvs/bin(/.*)	system_u:object_r:cvs_bin_t
/home/cvs/bin/cvs	system_u:object_r:cvs_exec_t
/home/cvs/bin/runcvs_[^/]*	system_u:object_r:cvs_wrapper_exec_t
/home/cvs/tmp(/.*)	system_u:object_r:cvs_tmp_t
/home/cvs/project[^/]*(/.*)	system_u:object_r:cvs_dir_t
/home/cvs/project[^/]*cvslock(/.*)	system_u:object_r:cvs_lockdir_t
/home/cvs/project[^/]*repository(/.*)	system_u:object_r:cvs_repository_t
/home/cvs/project[^/]*repository/CVSROOT(/.*)	system_u:object_r:cvs_cvsroot_t
/home/cvs/project[^/]*repository/CVSROOT/history	system_u:object_r:cvs_cvsroot_history_t
/home/cvs/project[^/]*repository/CVSROOT/val-tags	system_u:object_r:cvs_cvsroot_valtags_t

Der Security Context von SELinux wird über die CVS-Regeln mit “make relabel” oder “setfiles” gesetzt.

4 Verzeichnisse

4.1 CVSDIR/project_p1/cvslock

Die Sperrdateien werden in ein eigenes Verzeichnis gelegt, um Anwendern ohne Schreibberechtigung auf das Repository den Zugriff darauf zu erlauben. Dafür müssen diese Anwender Schreibzugriff auf das Verzeichnis der Sperrdateien erhalten. Dies wird automatisch über die Guppe “cvsp1user” erlaubt. Zusätzlich muß unter in CVSROOT/config der Eintrag LockDir passend gesetzt werden. Normalerweise werden diese Sperrdateien im selben Verzeichnis wie die Quelldateien erzeugt.

4.2 CVSDIR/tmp

Um das CVS so weit wie möglich vom System abzuschotten hat es ein eigenes Verzeichnis für temporäre Dateien. Die Einstellung für das temporäre Verzeichnis wird über die Option “-T VERZEICHNIS” den CVS-Server beim Programmstart, zum Beispiel durch den Wrapper, mit übergeben. Dir Gruppe cvsuser hat Vollzugriff auf dieses Verzeichnis.

4.3 CVSDIR/bin

Dieses Verzeichnis enthält den Wrapper zum Start des Servers und gleichzeitig den Server selbst. Über den Wrapper ist es möglich die Startoptionen und die Umgebungsvariablen des CVS-Servers hardcodiert festzuschreiben. Dadurch ist der Start mit immer gleichen Einstellungen gewährleistet und Manipulationsmöglichkeiten reduziert.

5 Konfiguration

5.1 SELinux

Der Regelsatz für die verwendeten wrapper und für das CVS befindet sich unter [18]. Durch die aktive Entwicklung der Policy kann es sein, daß Anpassungen an die aktuelle Policy erforderlich sind.

5.2 CVS

Grundsätzliche Informationen zu CVS befinden sich hier [5]. Ich möchte nicht weiter auf die eigentliche Konfiguration des Servers eingehen, denn dazu befinden sich zum Thema sichere Konfiguration von CVS-Servern

mehrere gute englischsprachige Artikel im Netz. Es sei kurz auf ein kleines Buch [8] verwiesen. Auch für Benutzer gibt es mehrere Veröffentlichungen im Internet. Es sei nur eine [14] erwähnt, ohne das dies eine Wertung darstellen soll.

5.3 CVSROOT/config

Die nachfolgende Zeile ist in der zugrunde liegenden Konfiguration die einzige Zeile:

```
LockDir=CVSDIR/project_p1/cvslock/
```

Sperrdateien werden durch diesen Eintrag im angegebenen Verzeichnis angelegt.

5.4 CVSROOT/logininfo

Hinweis außerhalb des Themas:

Diese Datei regelt die Handhabung von "cvs commit"-Meldungen. Auf dem Beispielsystem wird ein Perlskript aufgerufen, welches die Meldung formatiert und per Mail versendet. Der Regelsatz des CVS-Server enthält auch die passenden Einträge.

5.5 SSH

Auf dem Server kam eine aktuelle SSH-Version von [7] zum Einsatz. Damit der CVS-Zugang über SSH sicher funktioniert, ist als Startshell der CVS-Wrapper eingetragen. Dadurch können Benutzer von Windows und Linux indem sie CVS die SSH als Shell mit "CVS_RSH=ssh" angeben mit cvs, LinCVS [10] und WinCVS [9] einfach und sicher zugreifen. Unter Windows gibt es mehrere Wege zu einer Secure Shell zu kommen. Der erste ist über Cygwin [13]. Die nächsten zwei sind Pakete die ohne eine komplette Cygwin-Installation laufen und befinden sich hier [12, 11]. Dem Benutzer steht je nach Konfiguration des SSH-Servers eine Authentifizierung über ein persönliches Paßwort und/oder mit privaten Schlüssel zur Verfügung. Eventuell ist es notwendig eine zweite Instanz des SSH-Servers mit einer anderen Konfiguration an einem zweiten Port lauschen zu lassen, damit der Administrator einen paßwortgeschützten Zugang verwenden kann. Als weitere Konfigurationsmöglichkeit ist das Binden des vom Benutzer auszuführenden Befehls (meist eine Shell) an die individuellen Schlüssel. Mehr dazu steht in der Dokumentation der verwendeten Secure Shell.

5.6 CHROOT

SELinux mit seiner strikten Prozeßtrennung ersetzt in diesem Fall chroot. Ein Vorteil ist der geringere Aufwand bei neuen Programmversionen durch den entfallende Pflege der chroot-Käfige. Weitere Vorteile wären, daß man die Domäne im Gegensatz zum chroot-Käfig nicht ungewollt verlassen kann und der einfachere Aufbau des Servers.

5.7 CVS-Wrapper

Zwei statisch gelinkter Wrapper bieten auf diesem System die Möglichkeit die Startparameter der CVS-Servers administrativ festzuschreiben, ohne daß Benutzer diese ändern können. Dies gibt insbesondere für Benutzer mit SSH-Zugang. Der Wrapper ist durch den einfachen Aufbau und das statische Linken zum Beispiel mit dietlibc [6] nur 2k groß. Er startet den CVS-Server mit folgenden Optionen:

-allow-root=CVSDIR/project_p1/repository	spezifiziert die zulässigen CVSROOT-Verzeichnisse
-T CVSDIR/tmp	setzt das temporäre Verzeichnis
-f	benutzerspezifische Einstellungen in "~/cvsrc" werden ignoriert
server oder pserver	Betriebsart des CVS-Servers

Umgebungsvariablen:

CVSUMASK	007	Damit fehlen bei alle neu angelegten Dateien die Rechte für die Welt.
HOME	CVSDIR/project_p1/repository	Verzeichnis in welchen sich .cvsrc befindet. Dadruch soll das Durchsuchen der Benutzerverzeichnisse vermieden werden.
CVSROOT	CVSDIR/project_p1/repository	Voller Pfad zum Standard-Repository. Ist dieser nicht angegeben, muß das Repository via -d cvsroot bei jedem Befehl spezifiziert werden

Die Quellen des Programmes befinden sich unter [17].

6 Mögliche Verbesserungen

- Die Linuxberechtigungen sind ein Knackpunkt dieser Installation, denn sie werden erst durch das Zusammenspiel mit SELinux sicher. Ansonsten sind Änderungen an Stellen möglich, wo es nicht sein soll. Dieses Problem läßt sich mit ACLs einfach lösen.
- Ein intelligenter Wrapper, welcher abhängig vom Repository den security context des CVS-Servers ändert, wäre eine gute Lösung.

References

- [1] http://www.rz.uni-bayreuth.de/lehre/unix_rz/vorlesung/dateibaum/setuid.html
- [2] <http://www.idealx.com/en/doc/chrooted-ssh-cvs-server/chrooted-ssh-scv-server.html>
- [3] <http://www.idealx.com>
- [4] <http://www.nsa.gov/selinux/>
- [5] <http://www.cvshome.org>
- [6] <http://www.fefe.de/dietlibc/>
- [7] <http://www.openssh.org>
- [8] Gregor N. Purdy "CVS kurz&gut" O'Reilly ISBN-3-89721-229-3
- [9] <http://www.wincvs.org>
- [10] <http://www.lincvs.org>
- [11] <http://www.networksimplicity.com/openssh/>
- [12] <http://lexa.mckenna.edu&sshwindows/>
- [13] <http://www.cygwin.com>
- [14] <http://www.jfipa.org/publications/CVSGuide/>
- [15] ifrOSS Freie Lizenz für Texte und Textdatenbanken http://www.ifross.de/ifross_html/if1.html
- [16] Institut für Rechtsfragen der Freien und Open Source Software (ifrOSS) <http://www.ifross.de>
- [17] Quellen des Wrappers <http://www.securityenhancedlinux.de/>
- [18] CVS-Regeln für SELinux <http://www.securityenhancedlinux.de>
- [19] Shell-Skript "setcvsrighs" <http://www.securityenhancedlinux.de>