APPENDIX A

Converting from an Existing Installation to CiscoSecure

CiscoSecure UNIX Server software includes a conversion utility that enables you to convert an old configuration file into a new server control file and AA database file. The utility is called cnv and its syntax is as follows:

```
# /usr/local/etc/ciscosecure/bin/cnv old_CONFIG new_CONFIG
```

where *old_CONFIG* is the configuration file used with the public domain TACACS+ code. The file *new_CONFIG* is a single file that contains a new AA database file CONFIG.DB and a new control file CONFIG.FILE separated by a large bar. You will need to take the file *new_CONFIG* and create two separate files before loading *new_CONFIG* into CiscoSecure UNIX Server software.

The following display is an example of a configuration file for the public domain server:

```
accounting file = /var/tmp/accounting
default authentication = file passwd.1

user = lol {
    login = file /etc/passwd
}

user = user1 {
    service = exec {
       default attribute = permit
       acl=5
    }
}

user = user2 {
    # no exec configured, but commands are configured
```

```
    cmd = telnet {
permit 1.2.3.4
        deny .*
    }
}

user = user3 {
}

user = user4 {
    default service = permit
}

user = user5 {
    service = exec {
       autocmd=”telnet foo”
    }
}

user = user6 {
    before authorization “before.sh 0 $user $name $port”
}

user = user7 {
    before authorization “before.sh 1 $address $priv $method”
}

user = user8 {
    before authorization “before.sh 2 $type $service $status”
}

user = user9 {
    before authorization “before.sh 3 $address $name $port”
}

# After
user = user10 {
    after authorization “after.sh 0 ${user}@foo $status”
    service = ppp protocol = ipx {
acl=12
    }
}

user = user11 {
    after authorization “after.sh 1 $priv $status”
```

```
}

user = user12 {
    after authorization “after.sh 2 $method $type $service $status”
    service = ppp protocol = ip {
        acl=3
    }
}

user = user13 {
    after authorization “after.sh 3 $user $priv $method $type $service
$status”
}

user = user14 {
    after authorization “foobar.sh 3”
}


user = user15 {
    member= group15
}

group = group15 {
    after authorization “after.sh 2 $method $type $service $status”
    service = ppp protocol = ip {
        acl=7
    }

}

user = user16 {
    service=ppp protocol= lcp {
         timeout=50
    }
}


user = user17 {
    service=ppp protocol= ip {
         addr=1.2.3.4
         addr=3.4.5.6
         optional foo=a.b.c.d
         optional foo=e.f.g.h
    }
```

```
}


user = user18 {
    service=ppp protocol= ip {
        default attribute = permit
addr-pool=mci
    }
}


user=unexpired {
    default service = permit
    login = des gd1hIo7.oHKaY
    expires=”May 23 1999”

    service=ppp protocol=ipx {
addr=7.8.8.9
inacl=101
    }

    service=ppp protocol=ip {
        default attribute = permit
addr=1.2.3.4
addr=5.6.7.8
optional inacl=3
    }

    service=slip {
addr=1.1.1.1
inacl=101
optional outacl=102
optional outacl=103
outacl=104
    }

    service = exec {
        acl=4
    }

    cmd = telnet {
permit 131.108.13.111
permit 131.108.13.122
permit “131.108.13.124 /compress”
    }
```

```
    arap = cleartext “Arap secret 1”
    chap = cleartext “Chap secret 1”
}

user=expiring {
    login = des GeXwDbHn4sAhY
    expires=”Dec 24 1995”
}

user=expired {
    login = des jeUiHsLcWG6Vk
    expires=”May 23 1990”
}

user = foo {
    global = cleartext foobar
}

user = baz {
    login = cleartext bazbaz
}

user=gunexpired {
    member=unexpired_group
}

user=gexpiring {
    member=expiring_group
}

user=gexpired {
    member=expired_group
}

group=unexpired_group {
    # password = gunexpired
    login = des OfHc64b/xPzok
    expires=”Mar 5 1999”
    cmd = show {
        permit all.*
        deny .*
    }
}
```

```
group=expiring_group {
    #password = gexpiring
    login = des DgyEPacwmcSwQ
    expires=”Dec 24 1995”
}

group=expired_group {
    # password = gexpired
    login = des sgKPg47gsS0ho
    expires=”Mar 5 1991”
}

user=$enab5$ {
    # pass is $enab5$
    login = des T6q03EmzgvTec
}

user=$enable$ {
    # pass is $enable$
    login = des z8xJeOtQX5CnQ
}
```

The following display is the result of running the previous file through the CiscoSecure UNIX Server conversion utility:

```
default =  {
password = file “passwd.1”
member = WORLD
}

user = baz {
password = clear “bazbaz”
member = WORLD
}

user = expired {
password = des “jeUiHsLcWG6Vk”
expires = “May 23 1990”
member = WORLD
}

user = expiring {
password = des “GeXwDbHn4sAhY”
```

```
expires = “Dec 24 1995”
member = WORLD
}

user = foo {
password = clear “foobar”
password = pap “foobar”
password = chap “foobar”
password = arap “foobar”
member = WORLD
}

user = gexpired {
member = expired_group
}

user = gexpiring {
member = expiring_group
}

user = gunexpired {
member = unexpired_group
}

user = lol {
password = file “/etc/passwd”
member = WORLD
}

user = unexpired {
default service = permit
password = des “gd1hIo7.oHKaY”
expires = “May 23 1999”
password = arap “Arap secret 1”
password = chap “Chap secret 1”
member = WORLD
service = slip {
set addr = 1.1.1.1
set inacl = 101
set optional outacl = 102
set optional outacl = 103
set outacl = 104
}
service = shell {
default cmd = permit
```

```
set acl = 4
cmd = telnet {
permit 131.108.13.111
permit 131.108.13.122
permit “131.108.13.124 /compress”
}
}
service = ppp {
default protocol = permit
protocol = ipx {
set addr = 7.8.8.9
set inacl = 101
}
protocol = ip {
default attribute = permit
set addr = 1.2.3.4
set addr = 5.6.7.8
set optional inacl = 3
}
protocol = lcp {
}
}
}

user = user1 {
member = WORLD
service = shell {
default attribute = permit
set acl = 5
}
}

user = user10 {
#This should be converted to binary from shell script
#post-process = “after.sh 0 ${user}@foo $status”
member = WORLD
service = ppp {
protocol = ipx {
set acl = 12
}
protocol = lcp {
}
}
}
```

```
user = user11 {
#This should be converted to binary from shell script
#post-process = “after.sh 1 $priv $status”
member = WORLD
}

user = user12 {
#This should be converted to binary from shell script
#post-process = “after.sh 2 $method $type $service $status”
member = WORLD
service = ppp {
protocol = ip {
set acl = 3
}
protocol = lcp {
}
}
}

user = user13 {
#This should be converted to binary from shell script
#post-process = “after.sh 3 $user $priv $method $type $service $status”
member = WORLD
}

user = user14 {
#This should be converted to binary from shell script
#post-process = “foobar.sh 3”
member = WORLD
}

user = user15 {
member = group15
}

user = user16 {
member = WORLD
service = ppp {
protocol = lcp {
set timeout = 50
}
}
}

user = user17 {
```

```
member = WORLD
service = ppp {
protocol = ip {
set addr = 1.2.3.4
set addr = 3.4.5.6
set optional foo = a.b.c.d
set optional foo = e.f.g.h
}
protocol = lcp {
}
}
}

user = user18 {
member = WORLD
service = ppp {
protocol = ip {
default attribute = permit
set addr-pool = mci
}
protocol = lcp {
}
}
}

user = user2 {
# no exec configured, but commands are configured
member = WORLD
service = shell {
cmd = telnet {
permit 1.2.3.4
deny .*
}
}
}

user = user3 {
member = WORLD
}

user = user4 {
default service = permit
member = WORLD
}
```

```
user = user5 {
member = WORLD
service = shell {
set autocmd = “telnet foo”
}
}

user = user6 {
#This should be converted to binary from shell script
#pre-process = “before.sh 0 $user $name $port”
member = WORLD
}

user = user7 {
#This should be converted to binary from shell script
#pre-process = “before.sh 1 $address $priv $method”
member = WORLD
}

user = user8 {
#This should be converted to binary from shell script
#pre-process = “before.sh 2 $type $service $status”
member = WORLD
}

user = user9 {
#This should be converted to binary from shell script
#pre-process = “before.sh 3 $address $name $port”
member = WORLD
}

group = WORLD {
privilege = des “z8xJeOtQX5CnQ” 15
privilege = des “T6q03EmzgvTec” 5
}

group = expired_group {
# password = gexpired
password = des “sgKPg47gsS0ho”
expires = “Mar 5 1991”
member = WORLD
}

group = expiring_group {
#password = gexpiring
```

```
password = des “DgyEPacwmcSwQ”
expires = “Dec 24 1995”
member = WORLD
}

group = group15 {
#This should be converted to binary from shell script
#post-process = “after.sh 2 $method $type $service $status”
member = WORLD
service = ppp {
protocol = ip {
set acl = 7
}
protocol = lcp {
}
}
}

group = unexpired_group {
# password = gunexpired
password = des “OfHc64b/xPzok”
expires = “Mar 5 1999”
member = WORLD
service = shell {
cmd = show {
permit all.*
deny .*
}
}
}

-----------------------------------------------------


LIST config_aa_database_filename = { “CONFIG.DB” };
LIST config_license_key = { “061db8afcf66db981f3c” };
STRING config_accounting_database_filename = “/var/tmp/accounting”;
NUMBER config_logging_configuration = 0x7e;
NUMBER config_accounting_write_frequency = 5;

NAS config_nas_config = {
  {
    “”,/* NAS name */
    “arachnid”, /* secret key */
    “”,/* message catalog */
```

```
    2,/* username retries */
    2,/* password retries */
    1,/* is default key */
    1/* is trusted NAS for SENDPASS */
  }
};
```