Denis Makarov, 123RF.com

## Virus scanner and content filter with AD authentication

# THE CLEANER

You might want to reap the benefits of Active Directory's single sign-on for your virus scanning and content filtering. If you also use Squid to handle user access to the Internet, you have a front-row seat for "When worlds collide." **BY CHARLY KÜHNAST AND DANIEL VAN SOEST**

Many companies use a proxy to handle web access. These intermediaries on the network make it easy to look for viruses on pages accessed by users and in downloads. Company management often requires users to log in to the proxy for monitoring purposes, making it possible to assign individual user or group privileges, such as access to the intranet or extranet. But logging in means adding another username and another password – more things for users to remember – unless you happen to use the credentials of an existing system, such as a Windows domain controller to authenticate against the proxy. With a little support from the Samba project, the Squid proxy [1] can grant users access by referencing their Active Directory accounts.

## Part 1: Configuring Samba

Besides Squid, you will need Samba 3.x, Samba-Client, Samba-Winbind [2], and the Kerberos package, which is typically called *krb5*, or something similar, by most distributions. Once you have all of these components in place, you can launch into the configuration, starting with the Samba configuration file, *smb.conf*, which you will need to modify as shown in Listing 1.

Completing the Samba installation often modifies the second file, */etc/nsswitch.conf*, in one fell swoop. However, take a look anyway, to see if it looks like this:

```
passwd: files winbind
group: files winbind
```

Your next task is to add the server to the Windows domain:

```
net join -U domain administrator
```

After entering the administrative password, you should see the following success message:

```
Using short domain name - myworkgroup
Joined 'HOSTNAME' to realm ➥
        'example.com'
```

A tip: If your server refuses to join the domain, you should check the time on both the proxy and the domain controller. If they don't match, the Kerberos ticket system can easily trip over its own toes. A good idea is to set up an NTP daemon for all the systems involved.

## Checking DC Connectivity

This completes the Samba setup – you can now restart the Samba and Winbind daemons:

### The NTLM Authentication Mechanism

NTLM authentication (NT LAN Manager) [3] doesn't seem to be a particularly precise science. One example of this is found in the logs that will be full of type *TCP_DENIED/407* messages, although everything is working from the user's point of view. This collateral damage comes from the individual NTLM protocol login steps. The whole process that leads to these weird log entries is described in the Squid wiki [4]. The title, "The gory details," sums it up:

1. The client opens a connection to the proxy and issues a request without any authentication information. It repeats this for each request – although the usual approach would be to use authentication for a defined period of time.
2. The server replies with status code *407* and a *Proxy Authenticate: NTLM* Windows domain or further information. Another header could also point to other authentication mechanisms. A bug (or a feature) in Internet Explorer that fails to comply with RFC 2616 on this point requires all supported mechanisms to first state their NTLM or risk being ignored by IE.
3. Squid sets up the connection and forces the client to initiate a new one.
4. The client opens the connection and additionally passes in a header. The server responds with a *407* (proxy auth required) and resends the header *Proxy Authenticate NTLM Base64-encoded_challenge-package*. The TCP connection has to stay up after this.
5. The client sends a new GET request with a header of *Proxy-Authenticate: Base64-encoded_NTLM-response_to_challenge* including the username, the password (possibly in different encoding types), and the domain name.
6. If an error occurs, everybody goes back to square one. Otherwise, the proxy executes the GET command and doesn't ask again while the TCP connection is up.

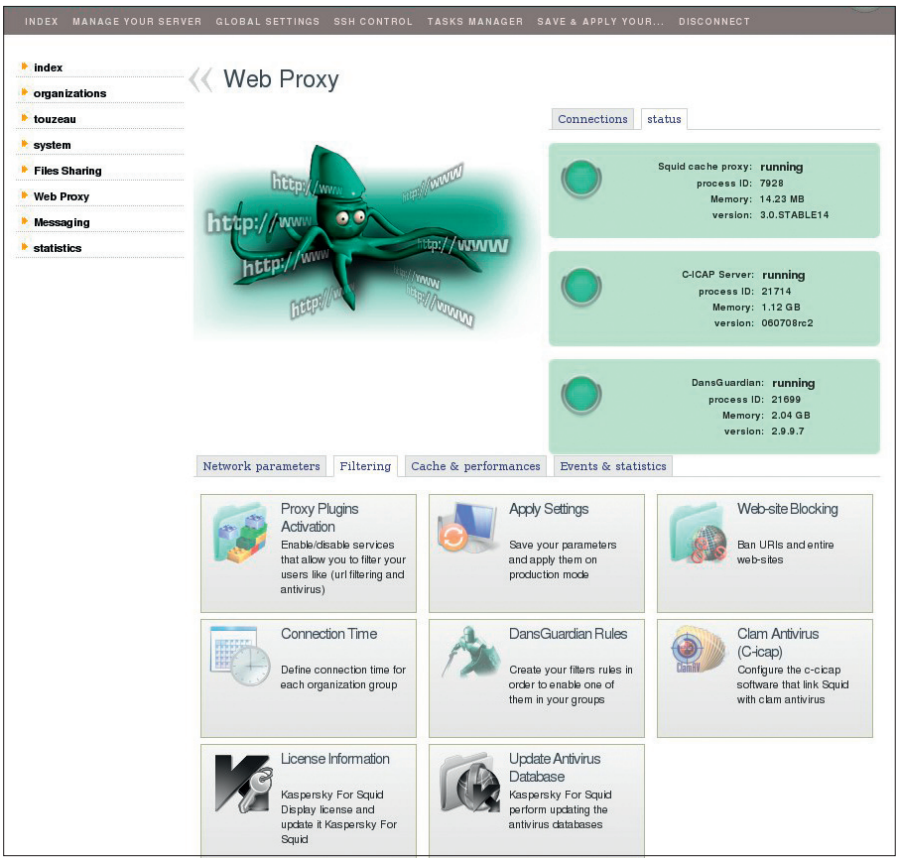


**Figure 1: DansGuardian provides a number of web proxy and content filtering services.**

## Listing 1: smb.conf

```
01 [global]
02 workgroup = myworkgroup    # Windows domain name
03 security = ADS
04 realm = example.com        # FQDN for the domain
05 password server = *        # Accept all password servers in the domain
06 encrypt passwords = true   # Encrypted password transmission
07 dns proxy = yes            # Retrieve domain data from DNS
08 idmap uid = 10000-20000    # Local reserved UID area for the domain users
09 idmap gid = 10000-20000    # Local reserved GID area for the domain groups
10 winbind separator = +      # How to separate the domain suffix from the user
   e.g.: myworkgroup+jones
11 [...]
```

## Listing 2: wbinfo_group.pl

```
01 # Original:
02
03 # $user =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("c",hex($1))/eg;
04 # # test for each group squid sent in its request
05 # foreach $group (@groups) {
06 # group =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("c",hex($1))/eg;
07
08 # After modification:
09
10 $user =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("U",hex($1))/eg;
11 # test for each group squid sent in its request
12 foreach $group (@groups) {
13 $group =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("U",hex($1))/eg;
```

```
/etc/init.d/smb restart
/etc/init.d/winbind start
```

Once these services are running, the following two commands

```
wbinfo -u
wbinfo -g
```

will return a list of all valid servers, systems, domain users (*-u*), and groups (*-g*). Then you can check to see whether the domain controller identifies you as a valid user with a defined group membership.

The tools you need for this – *ntlm_auth* and *wbinfo_group.pl* – are already in place on the system thanks to the packages you installed previously. Group names on the domain controller can contain non-standard characters.

To prevent Squid from tripping up when it encounters them, the proxy system must support UTF 8, and you need to modify slightly the script that was referred to previously, *wbinfo_group.pl* (Listing 2).

Check whether the system accepts the username (*jones* in our example) and the password (*mysecret*):

```
/usr/bin/ntlm_auth ➥
    -helper-Uprotocol=squid-2.5-basic
```

After pressing Enter, you can pass in the following to the tool, which actually isn't designed for this kind of user interaction:

```
myworkgroup+jones mysecret
```

If the connection to the domain controller works, it will respond with *OK*. Don't worry about the *squid-2.5* label; this only refers to the protocol – the example will work with more recent Squid versions (in fact, it will only work with them). Then you can check to see whether the setup identifies the user as a member of the *Internet* group:

```
/usr/sbin/wbinfo_group.pl
```

After pressing Enter, type the following at the command line:

```
myworkgroup+jones Internet
```

*wbinfo_group.pl* will respond with *OK*.

Later, you can remove the need to enter the domain name (*myworkgroup* in our example) by commenting out the *winbind separator = +* in *smb.conf* and setting the following:

```
winbind use default domain = yes
```

In future, if the user is already logged in to the domain, Internet Explorer or Firefox will pass their credentials directly to the proxy. Otherwise, a pop-up window appears prompting the user to enter a name and password.

At the end of the first section of the configuration, enter the following two commands

```
chmod 750 ➥
    /var/lib/samba/winbindd_privileged
chown root:nogroup ➥
    /var/lib/samba/winbindd_privileged
```

to modify the privileges for Samba on the filesystem.

## Part 2: Configuring Squid

What might have appeared to be an academic exercise in Part 1, now needs to be integrated with *squid.conf* as shown in Listing 3. You might have seen that *ntlm_auth* occurs twice with different helper protocols. Squid first tries with *ntlmssp* (SSP, Security Support Provider), transferring the encrypted username and password. If this step causes an error, Squid tries again (*basic*) and transfers the login information in the clear this time.

## More Granular Privileges Possible

To complete the basic configuration, add the lines from Listing 4 to the *ACL* section of *squid.conf*. At this point, you could add rules for more granular access on the basis of group membership.

For example, if you have two groups, one of which (*Everything*) is given unrestricted access, whereas the other (*Intranet*) is only allowed to surf the company's intranet, the configuration would look like Listing 5. The *internetusers* and *intranetusers* groups are defined to reflect this on the domain controller.

### Domain Free

To let users log in to the domain just with their usernames, you can comment out

```
winbind separator = +
```

in the *smb.conf* and then add the following line:

```
winbind use default domain = yes
```

If the user is already logged in to the domain, the credentials will be passed to the proxy directly by the browser. Otherwise a pop-up window appears, prompting the user to enter a username and password.

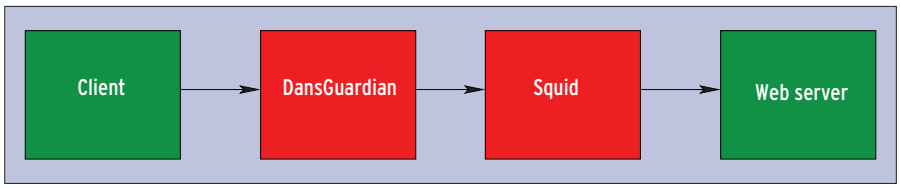


**Figure 2: The DansGuardian content filter is upstream of the Squid proxy for virus filtering.**

## Part 3: Virus Filter

If you additionally want the proxy system to handle virus checking, you can integrate a tool such as the DansGuardian content filter (Figure 1). DansGuardian has to be upstream of Squid from a logical viewpoint (Figure 2); this requires a couple of changes to the *squid.conf* file (Listing 6).

The first line tells the logfile to log the actual client IP address. Without it, you would just have the IP for the DansGuardian server, or just 127.0.0.1 if Squid and DansGuardian are running on the same hardware.

The second line serves a similar purpose, telling Squid to apply its ruleset to the "genuine" client IP. The user group and ACL definitions are in the next couple of lines.

After these preparations, the next step is to configure DansGuardian itself. It is important to build DansGuardian with the *--enable-ntlm = yes* compile option. To check easily whether this is the case, enter *dansguardian -v*. DansGuardian supports a large number of antivirus solutions; I will be using ClamAV for this example. Listing 7 shows you how to integrate DansGuardian.

### Listing 3: squid.conf (1)

```
01 auth_param ntlm program /usr/bin/ntlm_auth -helper-protocol=squid-2.5-ntlmssp
02 auth_param ntlm children 5
03 auth_param basic program /usr/bin/ntlm_auth -helper-protocol=squid-2.5-basic
04 auth_param basic children 5
05 auth_param ntlm max_challenge_reuses 0              # As of Squid 2.6
06 auth_param ntlm max_challenge_lifetime 60 minutes   # these lines are
07 auth_param ntlm use_ntlm_autonegotiate on           # not needed!
08 [...]
```

### Listing 4: squid.conf (2)

```
01 acl AuthUsers proxy_auth REQUIRED  # Require user authentication
02
03 http_access deny !AuthUsers        # Refuse non-authenticated users
04 http_access deny all               # Clean-up rule
```

### Listing 5: squid.conf (3)

```
01 acl whitelist.txt dstdomain "/etc/squid/whitelist.txt"
02 acl Everything external nt_group internetusers
03 acl Intranet external nt_group intranetusers
04 acl AuthUsers proxy_auth REQUIRED
05
06 http_access allow whitelist.txt Intranet
07 http_access allow Everything
08 http_access deny !AuthUsers
09 http_access deny all
```

Two user groups (defined by *filtergroups = 2*) each need a *dansguardianfn.conf* file. Both contain just a single line that defines the Active Directory group name specified in the Squid configuration. The files look like this:

```
[dansguardianf1.conf]
groupname = 'y1'
[dansguardianf2.conf]
groupname = 'y2'
```

To define group membership, you need to store the user to group mappings in */etc/dansguardian/lists/filtergroup*. It doesn't normally make much sense to edit this list manually; the *usermap.sh* script (which you can download [5]) provides an easier approach. The script was originally taken from the wiki on the DansGuardian website [6], but I made some changes, and it now runs nearly twice as fast as the original version. Of course, you will need to modify the group names in the top section of the script to match your needs. The */etc/dansguardian/lists/filtergroup* file shouldn't contain any tab stops because some systems replace these with line breaks during parsing, thus leading to incorrect group assignments.

Whenever the system reboots or the Winbind daemon is restarted, the privileges in the */var/run/samba/winbindd_privileged* file are reset. This means you'll need another script (*winbind-ch.sh*, which you can also download [3]) to correct the privileges. The following commands

```
mv winbind-ch.sh /etc/init.d/
update-rc.d winbind-ch.sh ➥
   start 21 2 3 4 5
```

run the script at system start time. Also, you will need to add it to your Winbind start script (Listing 8).

After restarting all the services, your virus checking proxy is now ready for action and will extract mail with test signatures from the network traffic. If you are of a cautious nature, you can improve ClamAV's detection rate by adding third-party patterns to the free virus scanner. Sanesecurity and MSRBL have a long history of providing reliable ClamAV signatures that are suitable for a variety of purposes. The focus is on detecting spam and phishing patterns, but you can put them to use on an HTTP proxy, too – if nothing else, they will prove valuable for web mailbox access.

The service is free; companies and larger organizations are asked for a donation on the website. Sanesecurity has a signature file with phishing patterns and another file for scams. MSRBL also provides two signature files, a generic antispam list, and a pattern list that identifies ads hidden in images.

Sanesecurity offers a shell script [7] that uses cron to update the signature files. It takes both the Sanesecurity and MSRBL signatures into consideration. When run, the script will wait for a period of between 30 seconds and 10 minutes to distribute the load more evenly over the mirror servers. Then it uses Rsync to write all the signature files that changed in the meantime to a temporary directory. The script then automatically checks whether ClamAV will accept the downloaded files before finally writing them to the ClamAV library directory and deleting the temporary files.

If you have enabled the *SelfCheck* option in your ClamAV configuration, it will integrate the new signatures the next time a check period ends. ■

## Listing 6: squid.conf

```
01 log_uses_indirect_client on
02 follow_x_forwarded_for allow all
03
04 acl usergruppe1 external nt_group y1
05 acl usergruppe2 external nt_group y2
06
07 http_access allow usergruppe1
08 http_access allow usergruppe2
09 http_access deny !AuthUser
10 http_access deny all
```

## Listing 7: DansGuardian Configuration

```
01 filterip = 127.0.0.1
02 filterport = 8080
03
04 proxyip = 127.0.0.1
05 proxyport = 3128
06
07 filtergroups = 2
08 filtergroupslist = '/etc/
   dansguardian/lists/filtergroupslist'
09
10 contentscanner = '/etc/dansguardian/
   contentscanners/clamav.conf'
11
12 authplugin = '/etc/dansguardian/
   authplugins/proxy-ntlm.conf'
13
14 daemonuser = 'dansguardian'  # Has to
   be identical to the clamav daemon user
15 daemongroup = 'nogroup'      # Has
   to be identical to the clamav daemon
   group sein!
```

## Listing 8: Changing the Privileges

```
01 case "$1" in
02        start)
03           log_daemon_msg "Starting
               the Winbind daemon"
               "winbind"
04              [...]
05           /etc/init.d/winbind-ch.sh
               start  # <-- privileges
               corrected
06           log_end_msg $?
07           ;;
08           [...]
```

## INFO


[1] Squid: *http://www.squid-cache.org*

[2] Samba: *http://www.samba.org*

[3] NTLM: *http://de.wikipedia.org/wiki/NTLM*

[4] Mechanics of NTLM authentication: *http://wiki.squid-cache.org/KnowledgeBase/NTLMAuthGoryDetails*

[5] Scripts for this article: *http://public.zii.krzn.de/dg-skripte/*

[6] DansGuardian: *http://dansguardian.org*

[7] Signature update script: *http://www.sanesecurity.co.uk/clamav/usage.htm*



**THE AUTHOR**

Charly Kühnast and Daniel van Soest are system and network administrators at the Lower Rhine Data Center, where they manage the centralized Internet infrastructure and live on bandwidth and disk space. In his leisure time, Charly loves cooking, and Daniel plays guitar with the punk rock band "4 dirty 5."