Practical Network Defense - Lab 8

Squid on ACME co.

Alessandro Serpi - 1647244

10 May 2019

Contents

1 Introduction

Squid is a caching web proxy that acts as an intermediary between clients and servers, it supports numerous protocols. In this laboratory we will set up Squid so to accept HTTP/S requests from hosts in the *client* network, forward them to the recipients and cache the relative answer. If an answer to a received request is already in cache, the request is is not forwarded and the cached resource is sent to the client.

Since LDAP authentication is required, it is not possible to create an intercepting (transparent) proxy: clients would believe they are communicating directly with the final receiver, however, they would receive proxy-related error status. Given that hosts in the *client* network can not request external resource using HTTP/S without passing through the proxy, a system-wide configuration is the most appropriate.

2 Squid installation

The assignment was carried out in a local environment.

A new VirtualBox guest running Arch Linux was created. Since the configuration was conducted through terminal commands, no graphical interface was added. Squid was installed using the default package manager (`pacman -S squid`) and started at boot time through *systemd* (`systemctl enable squid.service`).

The VirtualBox machine needs to be started passing the option `proxy` to the configuration script `start_vms.sh`. The executable powers up the necessary virtual machines (firewalls and domain controller) and configures the networks.

3 Forward proxy configuration

3.1 Squid configuration

Squid's configuration file is located at `/etc/squid/squid.conf`. Since a single Squid instance is started at a time, it is not necessary to create additional configuration files.

3.1.1 Root execution

In order to insulate the applicative from external interference, it is appropriate to create a new user `squid` for Squid. Since Squid is launched by a *systemd* service, it is necessary create a first statement to specify the user as which the proxy is executed. The second statement specifies the cache directory.

3.1.2 System setup

The second two statements allow only the local machine to access the cache manager.

`shutdown_lifetime` states for how much time the proxy must accept new connections after it is prompted to close (it can be done through `systemctl stop squid`). Since the applicative needs to be responsive, it is set up to a small value.

`visible_hostname` states the hostname that appears to clients when they use the proxy. It is the proxy's full domain name.

3.1.3 Forward proxy

`acl` statements are used to define access lists. Each list is composed by all requests that have in common the specified characteristic (port, source ip, etc.). Lists are allowed or denied specific actions with `http_access` statements.

`ssl_bump` statements specify the action associated with SSL/TLS connections for specific lists. In this case it is superfluous because `splice` (no bumb) is the default action.

The `http_port` statement specifies the port in which the proxy accepts incoming requests.

`refresh_pattern` statements specify custom thresholds for deciding for how long specific elements should be defined *fresh*. A non-fresh element is said to be *stale* and is automatically discarded.

3.2 Firewall rules

Remove all rules that allowed clients to navigate in internet and replace them with rules that allow HTTP/S traffic between clients and the web proxy. In addition, allow the proxy to surf the web.

To avoid locking out the administrators that have set the proxy in their machine, allow the web proxy to access the firewalls' web GUI (which uses port 443).

4 Authentication setup

Squid-provided `basic_ldap_auth` external program was adopted as the authentication plugin. It retrieves user credentials through basic HTTP authentication and forwards them to a LDAP domain controller. Therefore, it was necessary to insert a rule in both firewalls that allows LDAP traffic from the proxy to the domain controller.

Since anonymous searches are not allowed, a new computer LDAP account was created for the proxy using the domain controller's web GUI. `basic_ldap_auth` uses the aforementioned account (credential are provided with `-D` and `-w`) to search for (using the account's common name) and bind to the user-sent credentials. If the program replies with `OK`, the user is inserted in the `ldap-auth` group thanks to the instruction `acl ldap-auth proxy_auth REQUIRED`.

`auth_param basic children 10` states that Squid spawns at most ten instances of the authentication program. Since each instance can verify only one request at a time, if many users try to log in simultaneously, some of them may wait a few moments before being able to navigate.

5 Test of the setup

The IP address `100.64.2.5/24` (belonging to the client network) was assigned to a virtual interface of the host and the route `100.64.0.0/16` was added to the same interface via `100.64.2.1` (the internal firewall's address in the client network).

A Kathará guest was created in the client network. When it tried to connect to a web server (either in the private network or in the internet) trough *links*, no page could be reached.

In the host OS, a Mozilla Firefox instance was configured to use `100.64.6.3` as web proxy. When it tried to connect to a web server, the proxy requested user credentials. It was possible to navigate only if the inserted username and password corresponded to a valid LDAP user account.

6 Final remarks

In order to be effective, a caching system has to be installed in an environment where there are numerous close duplicate requests. Otherwise, if cache miss are much more frequent than hits, the caching mechanisms only slows down the connections.

Still, a Squid server has uses other than caching, such as centralised logging and traffic inspection.

7 Update: GUI access in the remote infrastructure

Although requirements specify that 'clients should not be allowed to navigate in HTTP/HTTPS without using the proxy', the next sentence states that 'HTTPS traffic should go straight to Internet', hinting that clients may access firewalls' and domain controller's web GUIs without using the proxy.

Furthermore, we believe that allowing access to the GUIs through the proxy poses a security risk. The proxy is located in the DMZ together with the web server, which is the main attack target because it is the only resource accessible from the outside. Since proxy and web server are located in the same collision domain, packets are not routed through the firewall-router. Therefore, an attacker that has compromised the web server has no restrictions in infiltrate the proxy, while he must use specific protocols to attack other hosts in the network.

If the proxy server has access to the firewall-routers and the domain controller, it becomes easier for an attacker to take control of the entire network. Therefore, we decided to allow only direct access to the GUIs, prohibiting connections through the proxy.