Lab 2 Forum: Questions & Answers

2 questions about Lab 2:

1. Should the server be used only as a proxy server or also as a web server, and show the HTML pages we wrote in Lab 1 in case the host is localhost?

2. Should we handle all additional requests that the browser sends, in case of browsers like Chrome and Firefox additional requests are sent with action-method we do not support, or alternatively HTTPS requests, therefore comes the question: are checks been made in a proxy server for requests whether they are legal or not, or should it transfer them as they are to the requested host without preforming a check in the proxy server?

1. Proxy server in the regular case and web server when one wants to see reports or edit the policy file. Don't return the HTML pages from Lab 1.

2. No need to support SSL (i.e. HTTPS), you support only HTTP

3. You need to transfer all requests (in HTTP of course) to the server and return answers.

4. No need to preform integrity (correctness) tests when you are proxy. Of course when one surfs to the log page or to the policy file editing page, then integrity is defiantly important.

HTTP 1.0 does not officially require a Host header but some applications (proxies) expect to see the Host header regardless of the protocol version.

as we learned in recitation in http/1.1 the header host was added.

Where exactly should we chceck for the "keyword" in case of http 1.0 request?

should we look for host header or should we check if the requested uri contains the keyword?

You should check the requested URI if it contains the keyword.

You must do that using the URI because, as you stated, hosts header is not mandatory in http/1.0.

Can you please upload some full http requests ( for each policy rule) that block because of some policy file.

You have such in the lab itself but I'll write one more here.

1. block-site "google.com"

a blocked request would look like:

GET http://www.google.com/ HTTP/1.0

2. block-resource ".exe"

a blocked request would look like:

GET http://awesome.game.com/downloadme.exe HTTP/1.1

host: game.com

3. block-ip-mask "194.90.42.216/24"

this must block any request in the range: 194.90.42.0-194.90.42.255 (194.90.42.216 is google's IP):

a blocked request would look like:

GET http://www.google.com/ HTTP/1.0

I am guessing we don't need to hand in a run.bat file for this lab2? Because it will simply print the usage.

You also wrote to hand in serverroot with our files...which I guess is copied from lab1.

Could you clarify what we need to hand in?

You should hand in run.bat with a command line that executes your server with your policy file.

Your server root is not necessaraly the same as in lab 1 because you have other files like policy editiing and log file.

In any way the index.html and params info from lab 1 are not relevant.

Is the "policy file" mandatory as a parameter when we call the server?

You said that the command line format is "java proxyServer [policy file]".

I'm not sure whether it means that "policy file" is optional or not.

The policy file is not optional.

In this lab we have no usage to the "defaultPage" value from the config file. Is that correct?

(I don't mind keeping it there, just want to make sure I got everything right).

Depends on how you design your logs and policy editor pages, but in general you're right

When we change policies using "http://content-proxy/policies", are these changes persistent?

i.e. if I add a rule, does it get saved to the original policies file? (therefore, if later I start the serve again, I will have the updated policies file)

Yes. It is persistent.

What files should be (if any) in "c:\serverroot"?

Both "policies" and "logs" page are generated dynamically from the server, and "config.ini", "policies.ini" and "output.log" are all exists in the same directory as the .java files of the program...

Therefore, the server root directory is empty...

One possibility is to create the policy and log HTMLs from an HTML file in the server root. the HTMLs does change dynamically, but the files are being used as templates. I think its an easier way to do that.

In any way, if you have no files in the serverroot, than submit an empty one.

Should we save the log file in the run directory or in serverroot directory (from the config file)?

The log path should be where the config.ini tells you to put it

Of course. But the value in the config.ini is relative, so my question is still valid.

e.g. if my run directory is "c:\run" and my root directory is "c:\serverroot" and the logPath value in the config.ini is "output.log".

Where should the log be?

A. "c:\run\output.log"

B. "c:\serverroot\output.log"

For me the first option makes more sense, but I would like to know what you expect

If your binaries are in "c:\run\" it does not mean that the current directory is "c:\run\" (i.e. "output.log" == "c:\run\output.log").

It all depends on how you execute the process (any directory can be the current directory).

Threrefore, the log path should be where the config.ini tells you to put it. If the given path is relative that just use the relative path. You can't tell up front where exactly it will be relative to your binaries simply because the current directory can be any directory.

Should we cross check policy rules?

For example:

Suppose the user types: "www.walla.co.il", and walla.co.il is listed as blocked site, thus the user receives "403 Access Denied" as expected.

The browser automatically requests www.walla.co.il/favicon.ico, which is a resource request, and suppose that .ico files are allowed (.ico extension is not listed in the blocked resources list).

In this case, should there be a cross checking of the policy rules?

If not, this contradicts that "walla.co.il" is blocked, isn't it?

If the user blocks walla.co.il, it blocks everything, including resources

Could you give an example how this page should look like?

HTTP request - what line should appear for that field? all lines? or just the first?

The same as you did in Lab 1.

HTTP request is the whole request header.

There wasn't any logs page in Lab 1...

Is it OK to implement the logs page as a large table in the following manner?:

Time Of Block: | Blocked HTTP Request: | Blocking Rule:

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

[DATE TIME] | [FULL HTTP REQUEST] | [BLOCKING RULE]

[DATE TIME] | [FULL HTTP REQUEST] | [BLOCKING RULE]

[DATE TIME] | [FULL HTTP REQUEST] | [BLOCKING RULE]

[DATE TIME] | [FULL HTTP REQUEST] | [BLOCKING RULE]

I wasn't talking about lab 1 logs, but lab 1 traces.

In any way, you can implement the log file in any way you want, as long as it is contains the information requested.

You wrote in the exercise: "Update config.ini with “logPath” entry. The entry specify a file that keeps the log of the proxy. logPath=”output.log” [submit with output.log]"

Do you mean that (1) we should keep "output.log" as the value in our config or (2) that we should submit the exercise with an example of the log file we got?

As in lab 1, you should submit the exercise with logPath="output.log".

Notice that we can change this value during the evaluation of the lab.

I'm not sure what do you expect us to do handle persistent connection over the proxy. For example, in the following scenario:

1. There is a request to http://www.google.com. This request goes through the proxy and the client get the response for its first http request.

2. There are some images in the page that some are located in other servers and some are located in the same server. Therefore the client generates more requests, some go to the server that the proxy has already established a connection with and some for other servers.

So what should we do in this case:

1. Should the proxy keep the persistent connection to www.google.com? (so there is no need to establish a connection to this host again)

2. If the answer to 1 was 'yes': How many persistent connection should we allow keeping (for each of the proxy's clients)?

In general it seems very complicated to support persistent connections through a proxy, if it is required I would ask you to explain what we should do (the desired behavior) in details, since otherwise I won't understand and I will have to ask this question again.

1. You can design and implement this as you like

2. I said in class that I one way to implement this is using persistent connections to the server, and I would keep persistent connection to the server as long as the browser keeps persistent connection to the proxy. You can bind these connections using some ID.

3. In the end, you can do (1). As long as it works correctly, I don't mind.

For clarification

there is policy & rule. One policy may contain many rules..

You say to "add/delete/modify each policy". As I understand you mean to add/delete/modify a rule.

Am I right?

Right.

IP/Mask policies: How does the checking should be done in this case?

The request does not contain the real IP reported by the DNS, should we send a ping request or something similar and perform AND operation with the IP and mask listed in the blocked IP addresses?

Or should we perform just a naive check of just matching the IP as is to the list of blocked IP address, (without bitwise operations)?

Before you perform the TCP connection to the website to the remote server, you should check if the IP is blocked or not.

Sure, but as I mentioned you could check in 2 ways:

1. The naive way of checking the blocked IP in case the user typed an IP address in the browser.

For example: the user entered: "http://192.168.10.12", thus the naive check looks up "192.168.10.12" in the blocked IP addresses list, in if a match found then the returned message is "403 Access Denied".

2. The smarter way is opening a connection socket just for retrieving the IP address from the DNS, and then close it immediately.

For example: the user entered: "http://www.walla.co.il", and suppose that the IP address of walla is "62.145.47.32" (and "walla.co.il" is not a blocked site), so from the connection socket this address can be extracted. After the extraction, the connection closes (without sending or receiving any data from remote target) and the check looks up the retrieved IP address in the blocked IP addressed list. Again, if a match found then the returned message is "403 Access Denied".

An extension to this option is to perform a bitwise AND operation on the extracted IP address with the items in the blocked addresses list considering their masks, which is of course more complex to implement.

What variation of the suggested solutions above should be implemented?

What are the exact requirements in Lab 2 for this check?

The "naive" approach is not "naive" but wrong.

If you block a certain IP it should be simply blocked. most of the times no user will surf to an IP, but to a DNS address. Before you TCP connect to the address (after DNS, of course) you need to check if it is block or not.

You still haven't related to the bitwise AND operation on the IP address with the items in the blocked addresses list considering their masks.

Should this check be implemented or not?

(which is of course more complex to implement)..

I don't mind how you implement this.

Can you please elaborate regarding:

1. block-site - what do you mean by host? any part of the request url?

2. block-resource - what do you mean by resource?

1. anything but the file name itself

2. file name itself

So in any case we should check the host in the URI. Even if we get 1.1, where "host" is part of the request we shouldn't rely on it?

in the example policy given you blocked "google.com/mail" - but if a user surfs to this specific URL we can't now if 'mail' is a file or a directory. if the policy filters a resource named 'mail' should it filter this request? should we treat this as a directory only if it ends with '/' in the end? or treat this as resource only if it has the pattern "\*.\*"?

Good point. So as you wrote, only if there's an extension treat that as a resource, otherwise it's a folder.