URL patterns

We want to be able to apply different rules and custom scripts for different websites. However, merely specifying "do this for https://example.com" is not enough. Single site's pages might differ strongly and require different custom scripts to be loaded. However, always matching against a full URL like https://example.com/something/somethingelse doesn't allow us to properly handle a site that serves similar pages for multiple values substituted for somethingelse.

Currently employed solution

Wildcards are being used to address the problem. Each payload and rule in Haketilo has a URL pattern that specifies to which internet pages it applies. A URL pattern can be as as simple as literal URL in which case it only matches itself. It can also contain wildcards in the form of one or more asterisks (*) that correspond to multiple possible strings occurring in that place.

Wildcards can appear in URL's domain and path that follows it. These 2 types of wildcards are handled separately.

Domain wildcards

A domain wildcard takes the form of one, two or three asterisks occurring in place of a single domain name segment at the beginning (left). Depending on the number of asterisks, the meaning is as follows:

- no asterisks (e.g. example.com) - match domain name exactly (e.g. example.com)
- one asterisk (e.g. *.example.com) - match all domains resulting from substituting * with a **single** segment (e.g. banana.example.com or pineapple.example.com but **not** pineapple.pen.example.com nor example.com)
- two asterisks (e.g. **.example.com) - match all domains resulting from substituting ** with **two or more** segments (e.g. monad.breakfast.example.com or pure.monad.breakfast.example.com but **not** cabalhell.example.com nor example.com)
- three asterisks (e.g. ***.example.com) - match all domains resulting from substituting *** with **zero or more** segments (e.g. hello.parkmeter.example.com or iliketrains.example.com or example.com)

Path wildcards

A path wildcard takes the form of one, two or three asterisks occurring in place of a single path segment at the end of path (right). Depending on the number of asterisks, the meaning is as follows:

- no asterisks (e.g. /joke/clowns) - match path exactly (e.g. /joke/clowns)
- one asterisk (e.g. /itscalled/*) - match all paths resulting from substituting * with a **single** segment (e.g. /itscalled/gnulinux or /itscalled/glamp but **not** /itscalled/ nor /itscalled/gnu/linux)
- two asterisks (e.g. /another/**) - match all paths resulting from substituting ** with **two or more** segments (e.g. /another/nsa/backdoor or /another/best/programming/language but **not** /another/apibreak nor /another)
- three asterisks (e.g. /mail/dmarc/***) - match all paths resulting from substituting *** with **zero or more** segments (e.g. /mail/dmarc/spf, /mail/dmarc or /mail/dmarc/dkim/failure but **not** /mail/)

If pattern ends **without** a trailing slash, it mathes paths with any number of trailing slashes, including zero. If pattern ends **with** a trailing slash, it only mathes paths with one or more trailing slashes. For example, /itscalled/* matches /itscalled/gnulinux, /itscalled/gnulinux/ and /itscalled/gnulinux// while /itscalled/*/ only matches /itscalled/gnulinux/ and /itscalled/gnulinux// out of those three.

If two patterns only differ by the presence of a trailing slash, pattern **with** a trailing slash is considered **more specific**.

Additionally, any path with literal trailing asterisks is matched by itself, even if such pattern would otherwise be treated as wildcard (e.g. /gobacktoxul/** matches /gobacktoxul/**). This is likely to change in the future and would best not be relied upon. Appending three additional asterisks to path pattern to represent literal asterisks is being considered.

Ideas for future additions

Right now a URL's query string is being completely disregarded for the purpose of matching the URL patterns. In the future, support for matching URLs with specific query parameters and maybe even HTTP(s) requests with specific POST parameters or specific cookies might me added.

Currently, protocols in the URL are matched exactly. However, as of Haketilo 3.0 support for protocol wildcard of http*:// is being developed which will match both "http://" and "https://".

The wildcards that have been added so far were designed to allow a reasonable level of flexibility, considering some common ways websites are served. However, only practice can show what works best, and so wildcard semantics are subject to change as the project matures.

Wildcard priorities and querying

In case multiple patterns match some URL, the more specific one is preferred. Specificity is considered as follows:

- If patterns only differ in the final path segment, the one with least wildcard asterisks in that segment if preferred.
- If patterns, besides the above, only differ in path length, one with longer path is preferred. Neither final wildcard segment nor trailing dashes account for path length.
- If patterns, besides the above, only differ in the initial domain segment, one with least wildcard asterisks in that segment is preferred.
- If patterns differ in domain length, one with longer domain is preferred. Initial wildcard segment does not account for domain length.

As an example, consider the URL http://settings.query.example.com/google/tries/destroy/adblockers//. Patterns matching it are, in the following order:

```
http://settings.query.example.com/google/tries/destroy/adblockers/
http://settings.query.example.com/google/tries/destroy/adblockers
http://settings.query.example.com/google/tries/destroy/adblockers/***/
http://settings.query.example.com/google/tries/destroy/adblockers/***
http://settings.query.example.com/google/tries/destroy/*/
http://settings.query.example.com/google/tries/destroy/*
http://settings.query.example.com/google/tries/destroy/***/
http://settings.query.example.com/google/tries/destroy/***
http://settings.query.example.com/google/tries/**/
http://settings.query.example.com/google/tries/**
http://settings.query.example.com/google/tries/***/
http://settings.query.example.com/google/tries/***
http://settings.query.example.com/google/**/
http://settings.query.example.com/google/**
http://settings.query.example.com/google/***/
http://settings.query.example.com/google/***
http://settings.query.example.com/**/
http://settings.query.example.com/**
http://settings.query.example.com/***/
http://settings.query.example.com/***
http://***.settings.query.example.com/google/tries/destroy/adblockers/
http://***.settings.query.example.com/google/tries/destroy/adblockers
http://***.settings.query.example.com/google/tries/destroy/adblockers/***/
http://***.settings.query.example.com/google/tries/destroy/adblockers/***
http://***.settings.query.example.com/google/tries/destroy/*/
http://***.settings.query.example.com/google/tries/destroy/*
http://***.settings.query.example.com/google/tries/destroy/***/
http://***.settings.query.example.com/google/tries/destroy/***
http://***.settings.query.example.com/google/tries/**/
http://***.settings.query.example.com/google/tries/**
http://***.settings.query.example.com/google/tries/***/
http://***.settings.query.example.com/google/tries/***
http://***.settings.query.example.com/google/**/
http://***.settings.query.example.com/google/**
http://***.settings.query.example.com/google/***/
http://***.settings.query.example.com/google/***
http://***.settings.query.example.com/**/
http://***.settings.query.example.com/**
http://***.settings.query.example.com/***/
http://***.settings.query.example.com/***
http://*.query.example.com/google/tries/destroy/adblockers/
http://*.query.example.com/google/tries/destroy/adblockers
http://*.query.example.com/google/tries/destroy/adblockers/***/
http://*.query.example.com/google/tries/destroy/adblockers/***
http://*.query.example.com/google/tries/destroy/*/
http://*.query.example.com/google/tries/destroy/*
http://*.query.example.com/google/tries/destroy/***/
http://*.query.example.com/google/tries/destroy/***
```

```
http://*.query.example.com/google/tries/**/
http://*.query.example.com/google/tries/**
http://*.query.example.com/google/tries/***/
http://*.query.example.com/google/tries/***
http://*.query.example.com/google/**/
http://*.query.example.com/google/**
http://*.query.example.com/google/***/
http://*.query.example.com/google/***
http://*.query.example.com/**/
http://*.query.example.com/**
http://*.query.example.com/***/
http://*.query.example.com/***
http://***.query.example.com/google/tries/destroy/adblockers/
http://***.query.example.com/google/tries/destroy/adblockers
http://***.query.example.com/google/tries/destroy/adblockers/***/
http://***.query.example.com/google/tries/destroy/adblockers/***
http://***.query.example.com/google/tries/destroy/*/
http://***.query.example.com/google/tries/destroy/*
http://***.query.example.com/google/tries/destroy/***/
http://***.query.example.com/google/tries/destroy/***
http://***.query.example.com/google/tries/**/
http://***.query.example.com/google/tries/**
http://***.query.example.com/google/tries/***/
http://***.query.example.com/google/tries/***
http://***.query.example.com/google/**/
http://***.query.example.com/google/**
http://***.query.example.com/google/***/
http://***.query.example.com/google/***
http://***.query.example.com/**/
http://***.query.example.com/**
http://***.query.example.com/***/
http://***.query.example.com/***
http://**.example.com/google/tries/destroy/adblockers/
http://**.example.com/google/tries/destroy/adblockers
http://**.example.com/google/tries/destroy/adblockers/***/
http://**.example.com/google/tries/destroy/adblockers/***
http://**.example.com/google/tries/destroy/*/
http://**.example.com/google/tries/destroy/*
http://**.example.com/google/tries/destroy/***/
http://**.example.com/google/tries/destroy/***
http://**.example.com/google/tries/**/
http://**.example.com/google/tries/**
http://**.example.com/google/tries/***/
http://**.example.com/google/tries/***
http://**.example.com/google/**/
http://**.example.com/google/**
http://**.example.com/google/***/
http://**.example.com/google/***
http://**.example.com/**/
http://**.example.com/**
http://**.example.com/***/
http://**.example.com/***
http://***.example.com/google/tries/destroy/adblockers/
http://***.example.com/google/tries/destroy/adblockers
http://***.example.com/google/tries/destroy/adblockers/***/
http://***.example.com/google/tries/destroy/adblockers/***
http://***.example.com/google/tries/destroy/*/
http://***.example.com/google/tries/destroy/*
http://***.example.com/google/tries/destroy/***/
http://***.example.com/google/tries/destroy/***
http://***.example.com/google/tries/**/
http://***.example.com/google/tries/**
http://***.example.com/google/tries/***/
http://***.example.com/google/tries/***
http://***.example.com/google/**/
http://***.example.com/google/**
http://***.example.com/google/***/
```

```
http://***.example.com/google/***
http://***.example.com/**/
http://***.example.com/**
http://***.example.com/***/
http://***.example.com/***
```

For a simpler URL like https://example.com the patterns would be:

```
https://example.com
https://example.com/***
https://***.example.com
https://***.example.com/***
```

Variants of those patterns with a trailing dash added would **not** match the URL.

Limits

In order to prevent some easy-to-conduct DoS attacks, older versions of Haketilo and Hydrilla limited the lengths of domain and path parts of processed URLs. This is no longer the case.

Alternative solution: mimicking web server mechanics

While wildcard patterns as presented give a lot of flexibility, they are not the only viable approach to specifying what URLs given settings of custom scripts should be applied to. In fact, wildcards are different from how the server side of a typical website decides what to return for a given URL request.

In a typical scenario, an HTTP server like Apache reads configuration files provided by its administrator and uses various virtual host, redirect, request rewrite, CGI, etc. instructions to decide how to handle given URL. It is possible using a schema that mimics the configuration options typically used with web servers would give more efficiency in specifying what page settings to apply when.

This approach may be considered in the future.