Using the Warhead Domain Configuration Tool

Alaric Snell

2001

Revision 1.0 10 May 2001 ABS

Revision 1.1 11 May 2001 ABS

The Basics

People who have domains hosted through Warhead get these basic facilities for each domain:

- DNS hosting
- Website hosting
- Email-domain forwarding

You get to control each of your domains using the domain-conf system. For example, imagine that you have `example.com` hosted by Warhead. You want to create a web site which can be referred to as either `www.example.com` or `example.com`. You want to create email addresses ending in `@example.com` and, seperately, `@lists.example.com`. You want a sepereate web site at `lists.example.com` to go with it.

The Warhead domain-conf system can let you set all that up!

A first look at domain-conf

Let's look at what we'd need to do so set up `example.com`.

Those two web sites need to be created first. Create a web site by making a subdirectory (with a meaningful name) under your home directory (you can access your home directory via ftp or ssh); in our example, we will call the two sites `example-main` and `example-lists`. Within each of those directories create a `public_html` directory for the actual Web site files, and a `warhead.conf` file containing:

```
[site]
email: webmaster@example.com
primaryname: www.example.com
```

...for `example-main/warhead.conf`, and:

```
[site]
email: webmaster@example.com
primaryname: lists.example.com
```

...for `example-lists/warhead.conf`.

The `email:` line lists the email address to list as being responsible for the site in error messages, and the `primaryname:` line tells the system what domain name to advertise that web site as normally being at (used when the same site can be accessed from more than one name, like `www.example.com` and `example.com`.

Having created our two web sites, we can now set up the domain control file. An initial domain control file will have been provided for you in your home directory under the name `warhead/`*`domain`* . In this case, it's `warhead/example.com`. Let's see what we need to put in there.

```
[domain]
last-modified: 2001010205         ; the date of our last update t
refresh: 86400                         ; These details do not nee
retry: 7200                         ; know what an SOA record is.
expire: 604800
minimum: 3600

[mail]                                         ; Email for addres
webmaster: example@warhead.org.uk ; webmaster@example.com redirec
*: example-com@hotmail.com         ; all other addresses go to ou

[web]                                         ; Web site at "http
site: example-main

[web.www]                         ; Web site at "http://www.examp
site: example-main

[web.lists]                         ; Web site at "http://lists.e
site: example-lists

[mail.lists]                         ; Email addresses ending "@l
staff: pete@aol.com, fred@foo.com, danny@things.net, example@warh
announce: me@me.com, me@myself.com, you@you.com, you@yourself.com
; Since there is no "*" rule, email to "fred@lists.example.com" w
; back to its sender.
```

Apart from the magic incantation at the top, the format of the file is actually quite obvious. Web sites contain the name of the web site directory, and email domains contain lists of email addresses and who they should go to, or the special address `*` to signal that emails to addresses that are not in the list should all be sent to one person instead of being bounced.

Note, in particular, that you can give a comma-seperate list of email addresses to send copies of incoming mail on to.

A detailed reference

Website directories

The domain-conf system finds a website directory in your home directory by being referred to it by name from a domain file.

The contents of a website directory are:

`public_html` This is compulsory. It must be a directory, and it is where the files you want to be available on your web site should be placed.

`cgi-bin` This directory is only required if you have set the `cgi:` flag in `warhead.conf`. It is mapped to `/cgi-bin/` in your web site, and will execute CGIs from the directory instead of serving up the contents of the file.

`error_log` This file will be created for you automatically when your site becomes activated. Each line in the file will be a description of a failed attempt to access part of your site. A lot of these happen naturally - don't worry about them.

`access_log` This file will be created for you automatically when your site becomes activated. Each line in the file will be a description of an attempt to access a page on your site. The defaul format is the Apache “combined” log format, and can be overridden in `warhead.conf`.

`warhead.conf` This file must follow the following format:

```
[site]
email: email address
primaryname: primaryname
logformat: Apache log format directive
directoryindex: default index.html file name

cgi: 1 or 0
```

The email address should be the webmaster of this site. It is compulsory. The primaryname is optional; if omitted, then one of the domain names pointing to this site will be picked at random to be the official “title” of the site. The log format string is an Apache logfile format string used to control the format of `access_log`. The directory index filename is the default filename to return when the browser requests the URL of a directory. If this file does not exist in the directory, a directory listing will be returned. The default is `index.html`. The cgi flag should be set to 1 if you have a `cgi-bin` directory for this site, and you want it to be enabled.

Domain files

The domain-conf system has a centrally maintained list stating which people have which domains. I control this. You need to ask me to put things in the list.

From that list, it finds out what domain files to read from whose home directories. Just putting a domain file in your home directory will not cause the system to do anything with it. I need to authorise the domain.

You will find a file for each domain you have inside the `warhead` directory under your home directory.

The file starts with a `[domain]` section. This always follows a standard format:

```
[domain]
last-modified: 2001010200
refresh: 86400
retry: 7200
expire: 604800
minimum: 3600
```

You will never need to modify these apart from `last-modified:`, unless you are a DNS expert and know what an SOA record is for. The last modified date needs to be changed whenever you alter the file; it goes `YYYYMMDDHH`, but since the domain system only updates once a day you might as well leave the `HH` as `00` and just update the date.

In the rare case of you having your DNS hosted by somebody else but your web or email stuff handled by Warhead, add `dns: 0` to the `[domain]` section to turn off Warhead DNS.

Then you have sections for each service you want at each domain. Service sections name the service and the domain address to provide that service at. For example, in a domain file for `example.com`, a section starting `[mail.foo]` will set up a mail domain with email addresses ending in `@foo.example.com`. A service with no domain name - such as `[mail]` - provides it's service to the master domain itself (`example.com` in our example).

`mail` services contain a list of redirection rules. Each rule looks like this:

```
peter: peter@yahoo.co.uk, peter@hotmail.com
```

In this example, email addressed to `peter@foo.example.com` ends up being delivered to `peter@yahoo.co.uk` and `peter@hotmail.com`.

If an email fails to match any of the rules, then a special rule called * is looked up. If it exists, then the email is sent to the addresses specified in that rule. If not, the email is returned to the sender.

web services are even simpler; they contain a single line, looking like:

```
site: example-main
```

This line names a web directory, as defined in the section called “Website directories”

For the intrepid, you can also cause raw DNS records to be inserted into your zonefile with rr services. They look like this:

```
[rr.anyUniqueLabelYouLike]
name: foo
type: A
body: 192.168.0.5
```

This causes your zone file to contain:

```
foo IN A 192.168.0.5
```

As with any BIND zone file, you can use absolute names, or @ to refer to the name of the zone:

```
[rr.anyUniqueLabelYouLike]
name: @
type: A
body: 192.168.0.5
```

```
[rr.anyUniqueLabelYouLike]
name: foo.bar.net.
type: A
body: 192.168.0.5
```

If you don't know what a zone file or a resource record is, you probably don't need to.

Raw zone files

Some of you will have elected to use raw zone files. These only configure DNS; they do not enable any Warhead email or Web site services.

Raw zone files also live in your warhead directory. They are BIND format zone files, with the name `domain.zone` . If you don't know about BIND format zone files, you will not be using raw zone files so I will not document their format here.

When the changes go live

Changing the domain-conf files has no immediate effect. Every night, the domain-conf program runs and collects everyone's domain files, then changes the actual configuration. That means that your changes will have no effect until about midnight.

If there is a problem with your domain configuration, the last working configuration is used and you get an email. If there is no problem, then your files get copied to a safe place; they are the “known working” version, which will be used if you make a change that breaks your version.

The astute reader will notice that, with some cunning, it's possible to do things that case Apache to refuse to start. Although I am working on making this harder to do by agressively checking things, please don't do it. This problem is not unique to domain-conf - under most Apache vhosting systems, renaming your `public_html` directory causes a big ruckus next time the Apache configuration gets reloaded. But I'm trusting you guys not to do it to me!