Subject: Re: Renegade FDS Running on Linux - on XWIS/WOL!!!
Posted by Blazer on Tue, 07 Mar 2006 15:24:50 GMT
View Forum Message <> Reply to Message

howang wrote on Tue, 07 March 2006 07:56Blazer, I think you should make it open-source. This is not RenGuard so you don't have a reason to make it closed-source, right?
mac tell me he love Open Source & GPL. Blazer, as a member of BHS, you should learn from your good friend, and make this open-source. This will help the Renegade Community.

Well since it's written in Perl, the source code is right there. XWISP currently only 100% emulates the protocol exchange of the windows FDS, so there is no reason to "modify" anything, as changing anything would make it no longer work.

Blazer wrote on Tue, 07 March 2006 13:40Terminator 101 wrote on Mon, 06 March 2006 19:16Question: Why Perl and not PHP? I think that PHP is more widely used.

PHP is more widely used for web applications, whereas perls main dominion is command line applications. Perl is well suited for things like this, and it's many built-in and addon functions for things like crc32 and other complex math, plus me being fairly adept at Perl, made it the logical choice. Pretty much any unix OS either has Perl installed by default or is available in the base package set, so I don't think anyone will have any problems running it.
howang wrote on Tue, 07 March 2006 07:56
I'm a PHP engineer. As what I know, Perl DO NOT have any built-in functions for complex encryption and maths function. What you can do is download modules for it. But on the other hand, PHP have dozen of built-in functions, such as sha1() , md5() and crc32() . These PHP functions are really BUILT-IN and you don't need to use it with any module.
PHP does have some nice built-in functions, along with Pear. Perl has vastly more modules available to do pretty much anything, from IRC modules to complex ANSI terminal emulation.

howang wrote on Tue, 07 March 2006 07:56
Your brenbot is packed with par, that means no perl installation is needed, so pre-installed perl or not is not a problem.

brenbot is packed with par so that it can run on Win32 systems. Since XWISP is for Linux, packing it will not be required. Starting it up will be as simple as ./xwisp.pl (And good thing too, since php does not have a par equivalent)

howang wrote on Tue, 07 March 2006 07:56I strongly suggest you write this in PHP. With PHP, you have easy access to the Internet without calling sockets directly. PHP will handle the socket for you. Any URL is accepted as filename and it allow us to treat URLs as normal files. For example, we can fopen('http://www.somewebsite.abc/file.ext','r'); in PHP! Can you do this in Perl? NO.
Yes, PHP does have builtin support to open and parse URLS (as it should, since it is a web programming language). But, for one thing, XWISP has no need to do this. But if I did, Perl does allow you to do the same thing, as I have used it for several other applications.

howang wrote on Tue, 07 March 2006 07:56PHP code is programmer-friendly and it have a good error reporting system, but perl do not.I am now starting to question if you know anything about Perl at all, because if you did, you would know that perl code and php code and syntax are very similar. And, with Perls "use warnings; use strict;" I would wager that it's error reporting is vastly superior to PHPs "Parse error on line 101"

howang wrote on Tue, 07 March 2006 07:56PHP is the next generation of scripting language, perl is outdated and unloved.LOL. Actualy most people would probably say that something like Ruby is the next generation scripting language. I suppose you could call Perl "outdated" in that it has been around for a long time, but it's definitely not unloved, or lacking any features, support, or development.

howang wrote on Tue, 07 March 2006 07:56As what you can see, this forum is using PHP-based script but not a Perl CGI. PHP is fast, stable and powerful, use PHP is much more better than Perl in any situation.Yes, as I said, PHP is well suited for web applications, such as this forum software. PHP was *designed* to be a web scripting language. Perl was designed to be a command-line scripting language. At one time Perl was tried to press into service as a web scripting language (mod_perl), which does work (still in use today for some applications), but php matured and developed to be mainly a web scriping/cgi language, with builtin support for most common web functions.

If you are indeed a Zend Certified PHP Engineer, you should agree with me on most if not all of the points I have made. There is no denying that PHP is made for web apps, and Perl is primarily designed as a console/command-line language. Now I'm not saying that PHP is incapable of doing command-line things (Crimson has done some pretty amazing things with PHP, including a renrem implementation as well as a RenGuard SSC), just that Perl is better suited to handle this sort of task.