Linux-Fu: Applications on the Web

Originalartikel

Backup

<html> <p>Did you know you can run remote Linux GUI programs in a browser with HTML5 support? It&#8217;s even secure because you can use SSH tunneling and that little trick means you don&#8217;t even need to open additional ports. If this sounds like gibberish, read on, it&#8217;s actually pretty easy to get up and running.</p> <p>I recently was a guest on <a href=„https://macrofab.com/blog/mep-ep57-mr-williams-your-book-changed-my-life/“ target=„_blank“>a Houston-based podcast</a>, and the hosts asked me if the best thing about writing for Hackaday was getting to work with the other Hackaday staff. I told them that was really good, but what I like best was interacting with people (well, most people) in the comments. That sometimes you&#8217;d post an article and someone would bring a topic&#160;up in comments that would really knock your socks off. This is how I wound up with this nearly ideal remote access solution, that requires nothing on the remote side but a web browser.</p> <p>A while back I posted about <a href=„https://hackaday.com/2017/03/10/linux-fu-keeping-things-running/“>keeping programs running after log off</a> on a Linux box. The post was mostly about non-GUI programs but you could use NX or VNC to handle it. In the comments, someone mentioned how unhappy they&#8217;d been with recent copies of NX and another commenter called [Screen for X11] posted about a tool called <a href=„https://xpra.org/“ target=„_blank“>xpra</a>.</p> <p/> <p>I had never heard of it, so I went to check it out. I was impressed. You could remote single applications (almost like doing an ssh -X; really nice if you are trying to use a little netbook into your massive desktop computer). You can also get a shadow copy of your normal desktop or create a new desktop. Performance was good and you could connect via ssh (or not), do certificate-based authentication, and more. Like many other similar solutions, you can exit and pick back up where you left off.</p> <p><a href=„https://hackadaycom.files.wordpress.com/2017/03/clock.png“ target=„_blank“><img data-attachment-id=„247841“ data-permalink=„http://hackaday.com/2017/03/31/linux-fu-applications-on-the-web/clock-30/“ data-orig-file=„https://hackadaycom.files.wordpress.com/2017/03/clock.png“ data-orig-size=„800,400“ data-comments-opened=„1“ data-image-meta=„{&quot;aperture&quot;:&quot;0&quot;,&quot;credit&quot;:&quot;&quot;,&quot;camera&quot;:&quot;&quot;,&quot;caption&quot;:&quot;&quot;,&quot;created_timestamp&quot;:&quot;0&quot;,&quot;copyright&quot;:&quot;&quot;,&quot;focal_length&quot;:&quot;0&quot;,&quot;iso&quot;:&quot;0&quot;,&quot;shutter_speed&quot;:&quot;0&quot;,&quot;title&quot;:&quot;&quot;,&quot;orientation&quot;:&quot;0&quot;}“ data-image-title=„clock“ data-image-description=„“ data-medium-file=„https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=400&amp;h=200“ data-large-file=„https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=800“ class=„alignright wp-image-247841 size-medium“ src=„https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=400&amp;h=200“ alt=„“ width=„400“ height=„200“ srcset=„https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=400&amp;h=200 400w, https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=250&amp;h=125 250w, https://hackadaycom.files.wordpress.com/2017/03/clock.png?w=768&amp;h=384 768w, https://hackadaycom.files.wordpress.com/2017/03/clock.png 800w“ sizes=„(max-width: 400px) 100vw, 400px“/></a>As I was reading the documentation, though, something caught my eye. There is actually an HTML5 client and web server built in. That means you can export applications from a Linux box to a web browser. This isn&#8217;t unique. There&#8217;s an Apache project called <a

href=„https://guacamole.incubator.apache.org/“ target=„_blank“>Guacamole</a> that will sort of do the same thing, but it requires a lot of overhead including a JSP server and performance wasn&#8217;t that great last time I tried it. Google has a remote desktop solution that uses Chrome, too. However, the xpra solution is pretty snappy and very flexible. You can see a screencast of me using xpra to a remote server in the video below. Here is a screenshot of a shell and a clock running inside Google chrome on a remote computer.</p> <h2>Without the Browser</h2> <p>The main emphasis seems to be using xpra as both a server and a client. You can find many <a href=„https://xpra.org/trac/wiki/Usage“ target=„_blank“>examples of common usage</a> on the project&#8217;s wiki. That&#8217;s a good place to start because the <a href=„https://xpra.org/trac/browser/xpra/trunk/src/man/xpra.1“ target=„_blank“>man page</a> reveals an enormous number of options. You can provide read-only access, export sound, share printers, link clipboards, scale video, and more &#8212; if you have the patience to wade through the documentation.</p> <p>There are versions for several operating systems, so even if you aren&#8217;t using Linux everywhere, you can still try it (honestly, though, I had some trouble getting the Windows GUI to work, although the command line was fine). I did find a version of xpra in the Ubuntu repositories, but it was out of date and didn&#8217;t want to work. It was much better to install from the project&#8217;s repositories.</p> <h2>Security</h2> <p>The biggest concern, of course, is security. You can set up xpra to use SSL authentication or passwords. You could also use a wrapper or <a href=„https://en.wikipedia.org/wiki/Port_knocking“ target=„_blank“>port knocking</a> to control access to a port. I decided to go a different route. I always have a firewall blocking ports that I don&#8217;t expect open on my Linux boxes. That way if something does try to open up a port I am not aware of it, it should break and in the process of fixing it, I will know what&#8217;s exposed to the outside world.</p> <p>What I did was let the xpra server run on an unused port locally, but I did not open the firewall for that port. In theory, then, anyone who could log into the machine could access the remote applications, but given this is a server with just a small number of administrators, they could all get into anything, anyway. Of course, remote access only on the same machine isn&#8217;t the point, right? That&#8217;s why I use an ssh tunnel to get to that remote port. Granted, that makes the convenience of the browser-based client a bit less, but then again, ssh clients that can create tunnels are widely available, so for my case that was acceptable and it seems relatively secure. If someone breaks into the server, they will have access to everything anyway so the remote access won&#8217;t really expose anything new.</p> <p><img class=„alignright“ title=„null“ src=„https://hackadaycom.files.wordpress.com/2017/03/xpralogin.png?w=333&amp;h=293“ alt=„xpralogin.png“ width=„333“ height=„293“/>You can see how I do the ssh tunnel in the video below. You can set up xpra to provide a login screen&#160;that (with a bit of configuration) will even work with SSL. The problem is, the web interface puts the password into the URL which means your passwords will be floating around in your browser history. Probably not a good idea.</p> <p>There&#8217;s not a one-size-fits-all solution to security. Before you expose your Linux box to the world, be sure you understand how someone could break into it and take steps to protect yourself.</p> <h2 style=„clear:none;“>In Use</h2> <p>It wouldn&#8217;t be hard to use this to provide read-only access in a web browser to an application running on a Raspberry Pi. This wouldn&#8217;t need&#160;to be read only, either if you were sure of your security. A common fallacy is to think you don&#8217;t care about security because your Raspberry Pi just does something simple. But if it is on the Internet, people may want to take it over as a platform to launch attacks on other people using your hardware and Internet connection. Don&#8217;t&#160;ignore any connected device, no matter how trivial, when it comes to security.</p> <p><iframe class=„youtube-player“ type=„text/html“ width=„800“ height=„480“ src=„http://www.youtube.com/embed/Ukqs5M5Fthg?version=3&amp;rel=1&amp;fs=1&amp;autohide=2&amp;showsearch=0&amp;showinfo=1&amp;iv_load_policy=1&amp;wmode=transparent“ allowfullscreen=„true“ style=„border:0;“>[embedded content]</iframe></p> </html>