UFW - Basics

Purpose

This article is to help provide some basic knowledge and information on setting up ufw for the ubuntu based computers.

Ubuntu is NOT as supported operating system for ECI. This page is simply here as a handy guide due to a request. If you are not comfortable managing your own system, we recommend using more user-friendly supported OS like Windows or Mac.

For the most part, most Linux flavor computers come with iptables as the firewall resource. ufw or Uncomplicated Firewall is the ubuntu equivalent, essentially it is iptables but simplified.

Requirements

You will need to have sudo or at least root privileges on the machine to execute iptables.

You should have console access to the machine in case you make a mistake with the firewall commands.

Basics

For the most part, here is a basic ufw baseline config you may wish to use, it is adapted from digital ocean firewall guide for ubuntu

You will likely have to modify this for your Ubuntu instance. This assumes you are blocking everything except for one IP address or range. i.e. ipaddress = 128.111.0.0/16 (campus wired network)

First make sure it is installed. If it is already installed, the below command will do nothing

```
sudo apt-get install ufw
```

At any time, you can check the status of UFW with this command:

```
sudo ufw status verbose
```

By default, UFW is disabled so you should see something like this:

```
Output:
Status: inactive
```

Set Up Default Policies

If you're just getting started with your firewall, the first rules to define are your default policies. These rules control how to handle traffic that does not explicitly match any other rules. By default, UFW is set to deny all incoming connections and allow all outgoing connections. This means anyone trying to reach your cloud server would not be able to connect, while any application within the server would be able to reach the outside world.

Let's set your UFW rules back to the defaults so we can be sure that you'll be able to follow along with this tutorial. To set the defaults used by UFW, use these commands:

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

As you might have guessed, these commands set the defaults to deny incoming and allow outgoing connections. These firewall defaults, by themselves, might suffice for a personal computer but servers typically need to respond to incoming requests from outside users. We'll look into that next.
sudo ufw default deny incoming
sudo ufw default allow outgoing

If we enabled our UFW firewall now, it would deny all incoming connections. This means that we will need to create rules that explicitly allow legitimate incoming connections—SSH or HTTP connections, for example—if we want our server to respond to those types of requests. You will probably want to allow incoming SSH connections so you can connect to and manage your machine.

This will create firewall rules that will allow all connections on port 22, which is the port that the SSH daemon listens on. UFW knows what "ssh", and a bunch of other service names, means because it's listed as a service that uses port 22 in the `/etc/services` file. We can actually write the equivalent rule by specifying the `port` instead of the service name. For example, this command works the same as the one above:

To configure your server to allow incoming SSH connections (world wide, example only, you will want to actually set some boundaries), you can use this UFW command:

<table>
<thead>
<tr>
<th>bad and good rule examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>This one is the BAD example, so DO NOT USE THIS ==&gt; sudo ufw allow ssh</td>
</tr>
<tr>
<td>Do NOT actually use the above command, if you do, you are basically allowing the entire planet to knock on your door. Use the command below to limit yourself to a specific range, for instance the campus wired network:</td>
</tr>
<tr>
<td>Good example, so USE THIS ==&gt; sudo ufw allow from 128.111.0.0/16 to any port 22</td>
</tr>
</tbody>
</table>

Remember, do NOT actually use the top command that does not specify a range, you will want to restrict your rules better

other common options include allowing web browsing, unlike the ssh options, internet rules restriction commands like `from 128.111.0.0/16 to any port` are optional:

http (unsecure internet): `sudo ufw allow http` or `sudo ufw allow 80`
https (secure internet): `sudo ufw allow https` or `sudo ufw allow 443`

Enable UFW

To enable UFW, use this command:

```
sudo ufw enable
```

You will receive a warning that says the "command may disrupt existing ssh connections." We already set up a firewall rule that allows SSH connections so it should be fine to continue. Respond to the prompt with `y`.

The firewall is now active. Feel free to run the `sudo ufw status verbose` command to see the rules that are set.

Deny Connections

If you haven't changed the default policy for incoming connections, UFW is configured to deny all incoming connections. Generally, this simplifies the process of creating a secure firewall policy by requiring you to create rules that explicitly allow specific ports and IP addresses through. However, sometimes you will want to deny specific connections based on the source IP address or subnet, perhaps because you know that your server is being attacked from there. Also, if you want change your default incoming policy to `allow` (which isn't recommended in the interest of security), you would need to create `deny` rules for any services or IP addresses that you don't want to allow connections for.

To write `deny` rules, you can use the commands that we described above except you need to replace "allow" with "deny".

For example to deny HTTP connections and force users to use secure browsing, you could use this command:

```
sudo ufw deny http
```

Or if you want to deny all connections from `15.15.15.51` you could use this command
sudo ufw deny from 15.15.15.5

Deleting Rules

Knowing how to delete firewall rules is just as important as knowing how to create them. There are two different ways to specify which rules to delete: by rule number or by the actual rule (similar to how the rules were specified when they were created). We’ll start with the delete by rule number method because it is easier, compared to writing the actual rules to delete, if you’re new to UFW.

By Rule Number

If you’re using the rule number to delete firewall rules, the first thing you’ll want to do is get a list of your firewall rules. The UFW status command has an option to display numbers next to each rule, as demonstrated here:

```
sudo ufw status numbered
```

Numbered Output:
Status: active

<table>
<thead>
<tr>
<th>To</th>
<th>Action</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>------</td>
<td>----</td>
</tr>
<tr>
<td>[1] 22</td>
<td>ALLOW IN</td>
<td>128.111.0.0/16</td>
</tr>
<tr>
<td>[2] 80</td>
<td>ALLOW IN</td>
<td>Anywhere</td>
</tr>
</tbody>
</table>

If we decide that we want to delete rule 2, the one that allows port 80 (HTTP) connections, we can specify it in a UFW delete command like this:

```
sudo ufw delete
```

Reset UFW Rules (optional)

If you already have UFW rules configured but you decide that you want to start over, you can use the reset command:

```
sudo ufw reset
```

There are a lot of online tutorials that cover the above, do not be afraid to ask questions. While Ubuntu is not a supported operating system, meaning you will have to do all this work yourself, ECI is more than happy to point you secure resources. We much rather solve issues before they happen rather than try to clean up stuff after the fact.

Related articles

- IPTable - Configuration Examples
- UFW - Basics
- Digital Piracy
- Securing Linux Computer
- I get ‘Critical Security Alert’ email from Google