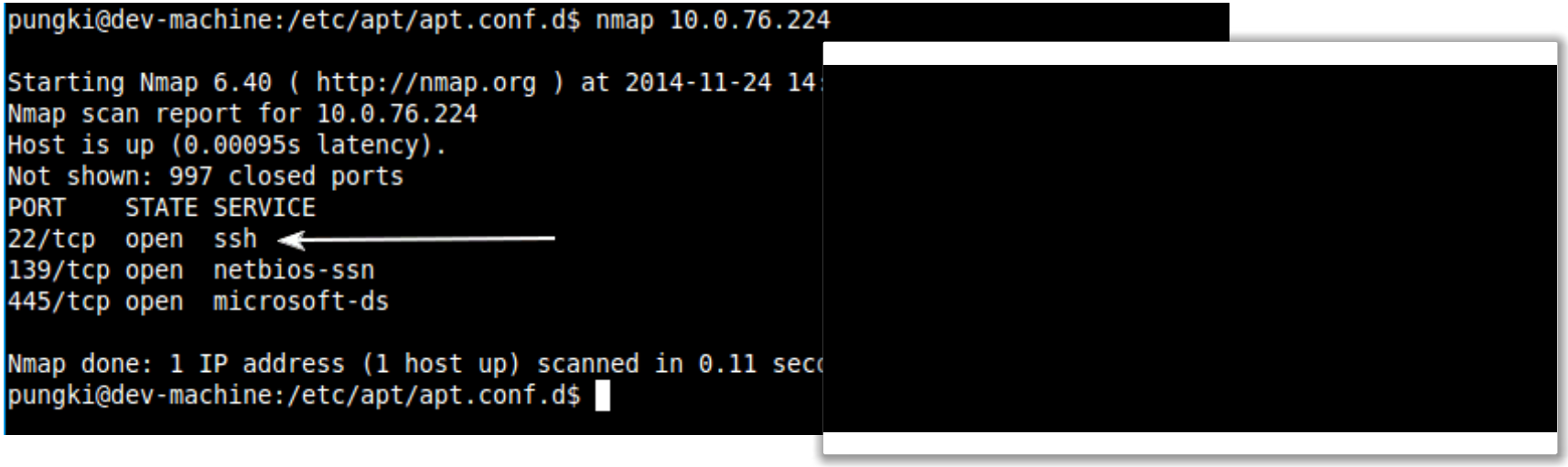
How to Hide Appplication Port Using knockd in Linux

Updated December 12, 2014 FIREWALL, LINUX IPTABLES, OPEN SOURCE TOOLS, SECURITY

As a system administrator, we should do everything to secure our server from attackers. As the internet grows, threats to our server is also growing. One of the popular entrances to attack our server is through the port on your server that open. If your SSH server is running on your machine, then usually the SSH port is listening. Which means it is open, waiting for the connection.

Leaving the port open for 24 hours, is not recommended because it is vulnerable. Because we can scan the machine to see the open port. **Nmap** is one of the most popular port scanner that can be used by anyone to scan your machine.



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

How if we can open the on demand and close the port when it's not used? Sounds interesting. Now we can do it using knockd application.

What is knockd

Knockd is a port-knock server. It listens to all traffice on an ethernet (or PPP) interface, looking for special "knock" sequences of port-hits. (Source : http://www.zeroflux.org/projects/knock)

How it works

Every application needs a port as a "door" for "listening" requests from other clients. This port usually on open state or close state. There are a lot of ports that available on the server. But there are some ports that agreed by consensus, such as SSH (22), Web (80) and FTP (21).

A basic rule of server security is to open only used ports and close the rest. You may have some ports that are sometimes used and sometimes not. Leaving those ports open while is not being used is not recommended.

When you install knockd, you can let the client "knock" the server with pattern. The knocking pattern will be unique to each other. If the pattern is match, then the port the request can enter your server.

Once you have done with the application, you can close the port manually or auto

How to install knockd

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

On this article, we are using Zorin 9 OS which based on Ubuntu 14.04 LTS. If you are using another distribution please adjust it to the installation method of your distribution.

Knockd is available on Ubuntu repository. Then we can use **apt-get** to install knockd.

```
$ sudo apt-get install knockd
```

Just wait for a few minutes, then your knockd is already setup.

```
Setting up knockd (0.5-3ubuntu1) ...
 * knockd disabled: not starting. To enable it edit /etc/default/knockd
Processing triggers for ureadahead (0.100.0-16) ...
pungki@dev-machine:/etc/apt/apt.conf.d$
```

Configure knockd

Knock configuration file is located in **/etc/knockd.conf**

The sample configuration is simple and easy to understand.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
[options]
        UseSysLog

[openSSH]
        sequence    = 1200,1300,1400
        seq_timeout = 10
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 1400,1300,1200
        seq_timeout = 10
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

We can see that the configuration is divided into three sections. The **[options]** section, **[openSSH]** section to open SSH port and **[closeSSH]** section to close SSH port.

By default **[options]** section contain only 1 row. It tell us that the knockd log will be recorded using the operating system log application. On Ubuntu, we will see the log in **/var/log/syslog**; folder.

Of course we can choose not to use SysLog. We can change it into this line, if we want to use custom log

```
logfile = /var/log/knockd.log
```

The above line, will put knock log file in **/var/log/knockd.log**

[openSSH] sections has identical commands with [closeSSH] section.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

sequence = 1200,1300,1400

This is the knock pattern. It will trigger the command below in the section. The value of this parameter is fully customized. We can choose another random number.

seq_timeout = 10

This will tell knockd about how long the knock pattern must be completed in.

command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

This parameter will open SSH port on port 22

tcpflags = syn

This parameter indicates that the Client will sends a TCP SYNchronize packet to the server

Setting up the firewall

As we know before, knockd will open particular port temporarily. So we have to make sure firewall is running on the server. Basically, we will close all ports. We are using iptables syntax to do it. Here are the steps.

```
pungki@dev-machine:~$ sudo /sbin/iptables -I INPUT -p tcp -m state
pungki@dev-machine:~$
pungki@dev-machine:~$
pungki@dev-machine:~$ sudo /sbin/iptables -I INPUT -p icmp -j ACCE
pungki@dev-machine:~$
pungki@dev-machine:~$
pungki@dev-machine:~$ sudo /sbin/iptables -A INPUT -j REJECT
pungki@dev-machine:~$
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

The 1st command, will allow the current on-going session through firewall.
The 2nd command, will allow the server is able to ping by another machine.
The 3rd command, will reject every requests.

To test the knockd service, we expect that our will firewall will drop all ssh connection. Then knockd will open it temporarily on demand.

```
pungki@dev-machine:/etc/apt/apt.conf.d$ sudo iptables -L |grep ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       udp  --  anywhere             anywhere             udp dpt:ssh
pungki@dev-machine:/etc/apt/apt.conf.d$
```

Test the knockd on server side

Once the firewall and knockd are setup, next we can test them.
To test the firewall, try remote the server via SSH from another machine. (*on this article, client IP is 10.1.6.14 and server IP 10.0.76.224*)

```
$ ssh -l pungki 10.0.76.224
```

With :
-l = login name
pungki = the user name on the destination server

If the firewall works, then we will get **“Connection refused”** error message.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
[pungki.arianto@web01 ~]$ ssh -l pungki 10.0.76.224
ssh: connect to host 10.0.76.224 port 22: Connection refused
[pungki.arianto@web01 ~]$
```

The reason we get the error message is the 10.1.6.14 is not allowed to enter the server. If we use this command, we will see no result.

```
$ sudo /sbin/iptables -L -n |grep 10.1.6.14
```

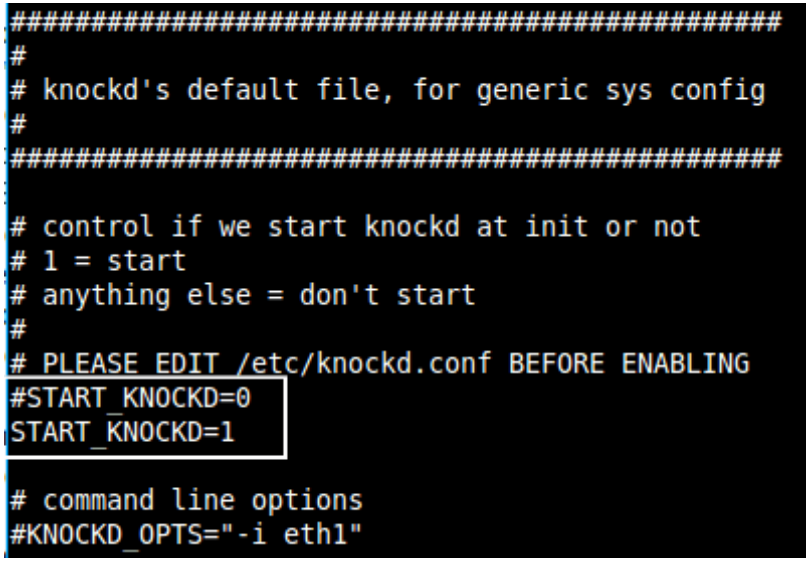
```
pungki@dev-machine:~$ sudo /sbin/iptables -L -n |grep 10.1.6.14
pungki@dev-machine:~$
```

Later, we will see the difference after knockd implemented.

Next step, is to test the knockd service.

To run knockd, we need to change knockd default file which located in /etc/default/knockd. Change the value of **START_KNOCKD parameter from 0 to 1**.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Save the file. Then type :

```
$ sudo service knockd start
```

**note : I tried to run the service using /etc/init.d/knockd start , but it always fail to sta*

Test knockd from Client side

On the client side, we need knock client to “knock” the server. On the client side, w

http://pkgs.repoforge.org/knock/knock-0.5.3.el5.rf.i386.rpm

Then run the command below to knock the server :

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
$ knock -v 10.0.76.224 1200 1300 1400
```

```
[pungki.arianto@web01 ~]$ knock -v 10.0.76.224 1200 1300 1400
hitting tcp 10.0.76.224:1200
hitting tcp 10.0.76.224:1300
hitting tcp 10.0.76.224:1400
[pungki.arianto@web01 ~]$
```

-v = verbose

10.0.76.224 = Server IP

1200 1300 1400 = knock sequence which defined in the knockd configuration

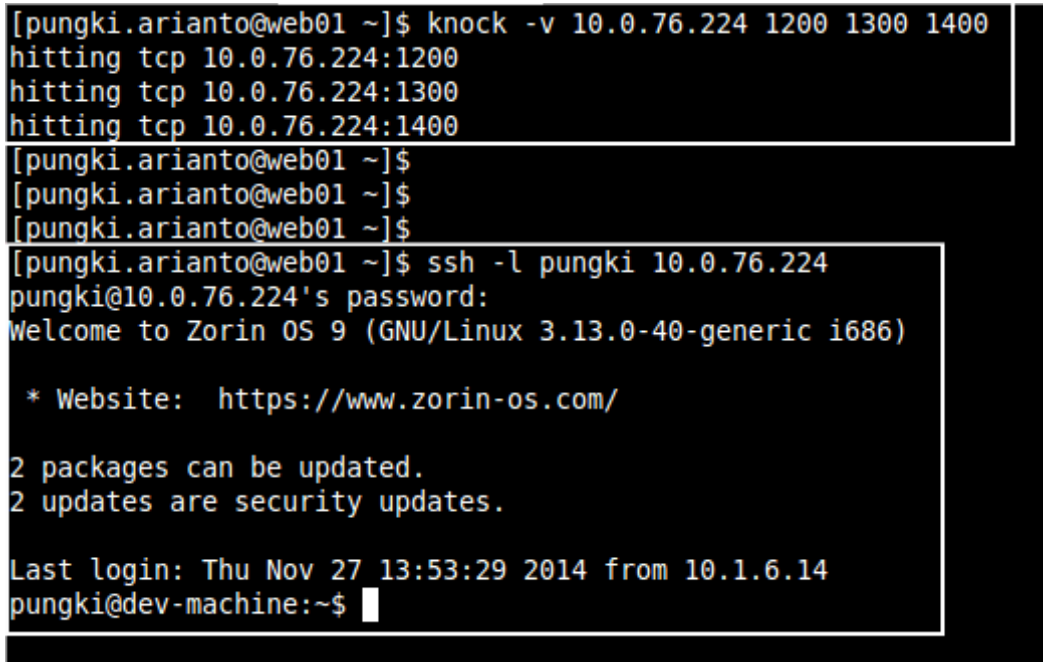
After knock the server, now we will see that the client IP is now allowed to enter the server.

```
$ sudo /sbin/iptables -L -n |grep 10.1.6.14
```

```
pungki@dev-machine:~$ sudo /sbin/iptables -L -n |gr
ACCEPT     tcp  --  10.1.6.14           0.0.0.0/0
pungki@dev-machine:~$
```

Then we can run SSH to remote the server.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



As we can see on the above image, the host name of are different. After SSH to the remote machine established then the host name is changed from @web01 into @dev-machine.

Close the port

After the client done to remote the server, the client need to close the port. To do t

```
$ knock -v 10.0.76.224 1400 1300 1200
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
[pungki.arianto@web01 ~]$ knock -v 10.0.76.224 1400 1300 1200
hitting tcp 10.0.76.224:1400
hitting tcp 10.0.76.224:1300
hitting tcp 10.0.76.224:1200
[pungki.arianto@web01 ~]$
```

***Please be careful**, to close the port, we put the knock sequence in the opposite order.*

After knock the server, we will see a prompt again. To check if the knock was success, we use iptables command again. If it was success, we will see that IP 10.1.6.14 will disappeared.

```
pungki@dev-machine:~$ sudo /sbin/iptables -L -n |grep 10.1.6.14
ACCEPT     tcp  --  10.1.6.14           0.0.0.0/0           tcp dpt:22
pungki@dev-machine:~$
pungki@dev-machine:~$
pungki@dev-machine:~$
pungki@dev-machine:~$ sudo /sbin/iptables -L -n |grep 10.1.6.14
pungki@dev-machine:~$
```

On the previous one, after knock to open the port, we saw that the client IP Address - 10.1.6.14 - is allowed to enter the server by firewall. Now, after we knock to close the port, if we check with the same iptables command, the rule was deleted.

Close the port Automatically

Since the close port activity is triggered by client, we will have the possibility that t
happen. So we can configure knockd to close the port automatically.

In order to do that, **we need to customize** the knockd configuration file. Here is th

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
[options]
        UseSysLog

[open_close_SSH]
        sequence    = 1200,1300,1400
        seq_timeout = 15
        start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        cmd_timeout = 10
        stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

The command still looks identical. The difference with the previous configuration is we put [openSSH] section and [closeSSH] section in the same block.

Then we add **cmd_timeout = 10** line to tell the server **to execute the stop_command 10 seconds after start_command is executed**. The port will be automatically closed, but the established connection remain connected.

Conclusion

Knockd help us to minimize the risk of leaving all ports open all the time. With knockd, we can open ports we need on demand. To improve security, we need to know the knock sequence before be open the port.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

SHARE ON FACEBOOK GOOGLE+ TWITTER PINTEREST LINKEDIN REDDIT

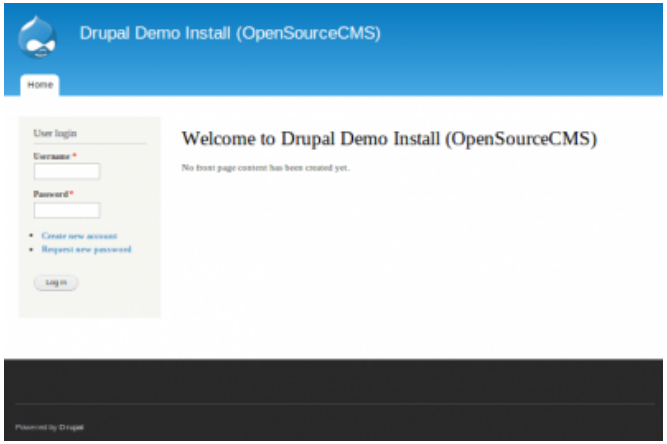
About Pungki Arianto

Pungki , currently working as a Linux / Unix administrator for a banking company. H
since it's fun for him. He is also interested in information technology, information se

Like to become part of Linoxide Team and contribute tips? Conta

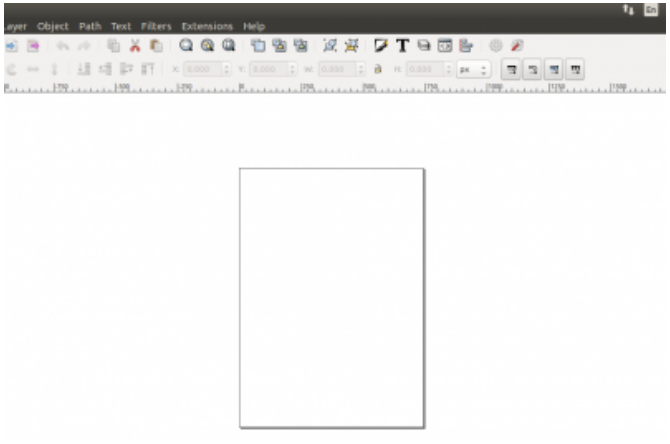
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Hand-picked related articles



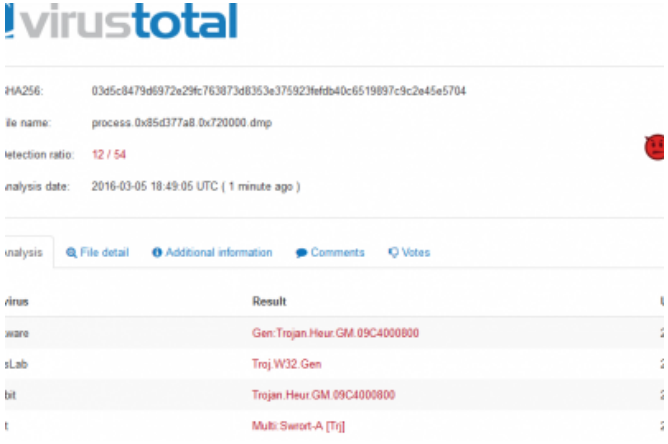
How to Install Drupal 7.x with Apache 2.x on Ubuntu 14.04

Drupal is an open source software maintained and



Install Inkscape - Open Source Vector Graphic Editor

Inkscape is an open source vector graphic editing



How to Setup Volatility Tool for Memory Analysis

In the IT security field, memory or Random Access Memory (RAM) analysis helps to identify the

Comments

Your email address will not be published. Required fields are marked *

Name *

Email *

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

All comments are subject to moderation.

What is linoxide based on ? Windows or Linux ? Type your answer into the box

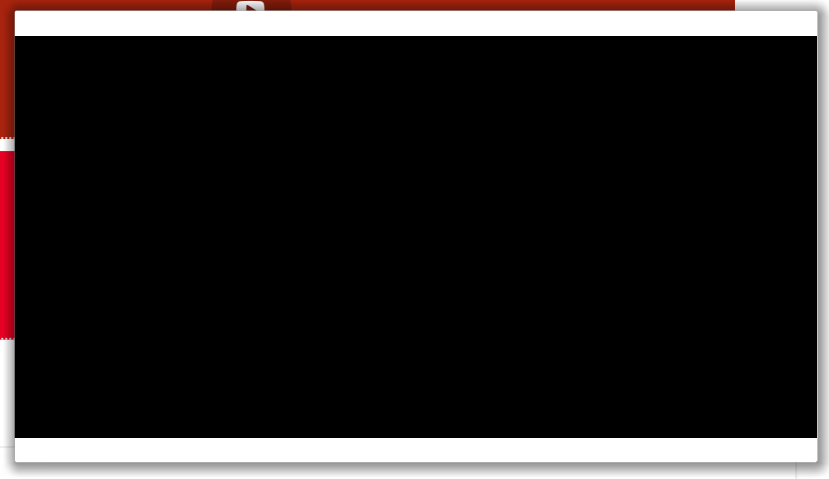
Submit Your Comment

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

JOIN OUR 16.9K LINUX COMMUNITY

Twitter

Facebook



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Search Linoxide Tutorials

LATEST GIVEAWAYS!

How to Install Google Chrome on Debian

How to Add a User to Sudoers on Debian

How to Configure sources.list on Debian 10

Linux Commands Cheat Sheet

Download PDF Version

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

Beginners Kubernetes Practical Guide

Download Your Free Ebook

Enjoyed the Read?

Don’t miss our next article!

Enter Your Email *

SUBSCRIBE

100% privacy – We will never spam you

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

ABOUT & CONTACT PAGES

Notify Any Errors

Provide News & Tips

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

JOIN OUR 16.9K LINUX COMMUNITY

Ask & Share – Join Our Facebook Community

SUBSCRIBE TO FREE NEWSLETTER

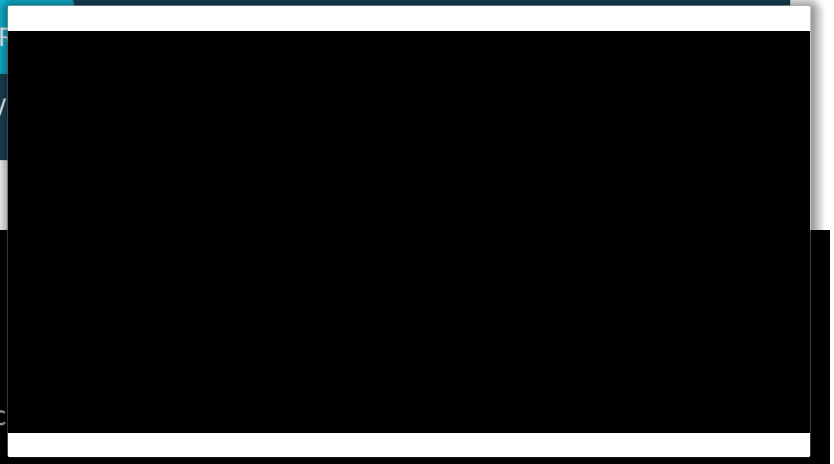
Enjoyed the Read?

Don't miss our next article!

Enter Your Email *

SUBSCR

100% privacy – We will never spam y



TERMS OF SERVICE / PRIVACY / CONTAC

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

COPYRIGHT © 2019 BTREME. ALL RIGHTS RESERVED

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD