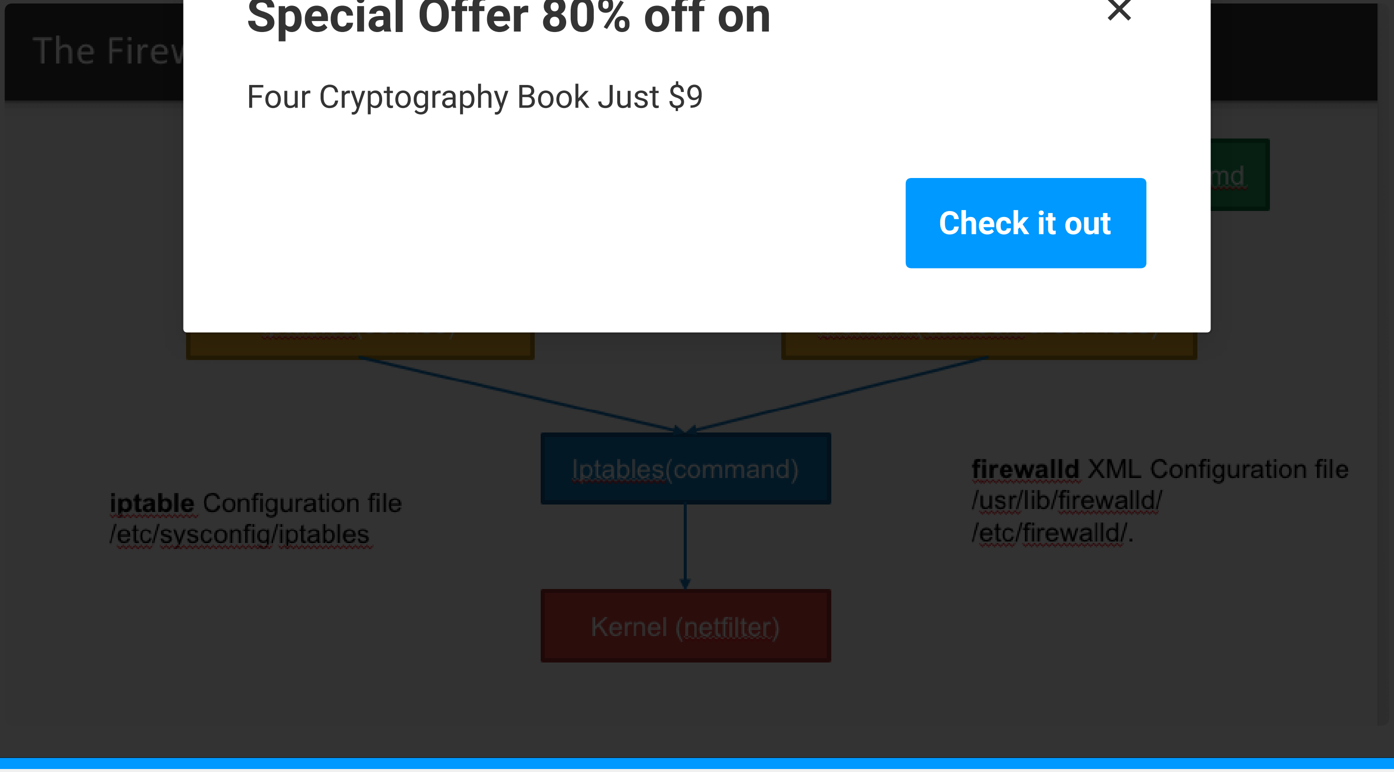
iptables

by Anish

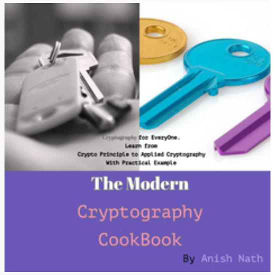
Posted on Saturday August 18



Introduction

This sample chapter extracted from the book, The Modern Cryptograhy CookBook . The Book theme is Cryptography is for EveryOne. Learn from Crypto Principle to Applied Cryptography With Practical Example

Get this book on Just $9 by availing coupon discount



The Modern Cryptography CookBook

Cryptography is for EveryOne. Learn...

Anish Nath

Buy Now

Minimum price: $14.99
Suggested price: $24.99

IPtables is the firewall service that is available in a lot of different Linux Distributions. While how easy it is to use and how quickly you can be on your way mucking around with your firewall

iptables CHAINS

Iptables is made up of 5 tables, each associated to specific functionalities of the net filter and each split into several "chains", specifying the functionalities of each table further

- **INPUT** - Used to control the behavior of INCOMING connections.
- **FORWARD** - Used to control the behavior of connections that aren't delivered locally but sent immediately out.
- **OUTPUT** - Used to control the behavior of OUTGOING connections.
- **PREROUTING**: This chain is used to make any routing related decisions before (**PRE**) sending any packets. Here is an example, we are redirecting any traffic that just reached the server on port 80 to the port 8080:

  ```
  iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-po
  ```

- **FORWARD**: As the name suggests, The `FORWARD` chain of `FILTER` table is used to forward the packets from a source to a destination. Here is an example of `FORWARD` chain where any `TCP` traffic received on port 80 on interface `eth0` meant for the host `192.168.0.4` will be accepted and forwarded to `192.168.0.4`:

  ```
  iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 192.168.0.4 -j ACCEPT
  ```

iptables Actions

- **ACCEPT**: Allow the connection
- **DROP**: Drop the connection (as if no connection was ever made; Useful if you want the system to disappear on the network)
- **REJECT**: Dont allow the connection but send an error back.

iptables Default Policy

In every linux system, the chain is configured with default ACTION, in order to know what is the default policy

```
sudo iptables -L | grep policy
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

How to Change Default iptables Policy

sysadmins can change the default policy by `iptables --policy <CHAIN> <ACTCION>`

for example

```
iptables --policy INPUT DROP
iptables --policy OUTPUT ACCEPT
iptables --policy FORWARD DROP
```

Get familiar you self with iptables rules `iptables -h` , this is great place to start, some tips

- `iptables -A` will add the rule at the end
- `iptables -I` will add the rule at the top by default
- `iptables -D` will delete a rule (specify a rule number or specify the whole rule you want to remove for this option to work)
- `iptables -C` will check for the existence of a rule
- `iptables -F` Delete all rules in chain or all chains

Most common IPtables rules

- **iptables: How to Block All Traffics**

```
iptables -F
iptables -A INPUT -j REJECT
iptables -A OUTPUT -j REJECT
iptables -A FORWARD -j REJECT
```

- **iptables How to Block Incoming Traffic Only**

```
iptables -F INPUT
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -j REJECT
```

- **iptables How Block Outgoing Traffic Only**

```
iptables -F OUTPUT
iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -j REJECT
```

- **iptables: How to Block Specific Incoming port or Service**
  This will block http service any incoming traffic

  ```
  iptables -A INPUT -p tcp --dport 80 -j REJECT
  ```

  or

  ```
  iptables -A INPUT -p tcp --dport www -j REJECT
  ```

  to allow only local interfaces for http

  ```
  iptables -A INPUT -p tcp --dport 80 -j REJECT
  ```

- **iptables: How to block specific host**

  This will block all access by that host

  ```
  iptables -A INPUT -s <remote_ip> -j REJECT
  ```

- **iptables: How to block outgoing to specific hosts**

  ```
  iptables -A INPUT -s <remote_ip> -j REJECT
  ```

- **iptables: How to allow access to specific mac address only**

```
iptables -A INPUT -m mac --mac-source <mac_address> -j ACCEPT
iptables -A INPUT -j REJECT
```

- **iptables: How to allow only SSH**

```
iptables -A INPUT -j REJECT
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -j REJECT
```

- **iptables: How to block all outgoing connection for example telnet**

```
iptables -A OUTPUT -p tcp --dport telnet -j REJECT
```

- **iptables: How to block ping**

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

or

```
iptables -A INPUT -p icmp --icmp-type 8 -j DROP
```

- **iptables: How to configure connection wait**

Makes iptables wait 15 seconds between new connections from the same IP on port 22 (SSH):

```
iptables -A INPUT -p tcp -i eth0 -m state --state NEW --dport 22 -m recent
```

- **iptables: How to Block Smurf attacks**

```
iptables -A INPUT -p icmp -m icmp --icmp-type address-mask-request -j DROP
iptables -A INPUT -p icmp -m icmp --icmp-type timestamp-request -j DROP
iptables -A INPUT -p icmp -m icmp -j DROP
```

- **iptables: How to drop excessive RST packets to avoid smurf attacks**

```
iptables -A INPUT -p tcp -m tcp --tcp-flags RST RST -m limit --limit 2/seco
```

- **iptables: How to do Port Forwarding**

  This rules will forward all the incoming request on port 80 to port 8080

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

  This rules will forward all the incoming request on port 80 from localhost to port 8080

```
iptables -t nat -I OUTPUT -p tcp -d 127.0.0.1 --dport 80 -j REDIRECT --to-p
```

- **iptables How to List IPtables Rules**

```
iptables -L
iptables -t nat --line-numbers -n -L
```

```
Ubuntu: sudo /sbin/iptables-save

RedHat / Centos: /sbin/service iptables save

Others: /etc/init.d/iptables save

Generic:  iptables-save > /etc/sysconfig/iptables
```

- **How to restore iptables rules from file**

```
  sudo iptables-save | sudo tee /etc/iptables.conf
  sudo iptables-restore < /etc/iptables.conf
```

- **How to flush clear all iptables rules**

  This command will not clear NAT rules

```
iptables -F
```

  Note if there are NAT rule, then to flush it

```
iptables -t nat -F
```

- **iptables: How to delete PREROUTING NAT rule**

  First find out what line it is by `iptables -t nat -L --line-numbers`

```
iptables -t nat -L --line-numbers
Chain PREROUTING (policy ACCEPT)
```

```
2    REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:

Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REDIRECT   tcp  --  anywhere             localhost            tcp dpt:
2    REDIRECT   tcp  --  anywhere             localhost            tcp dpt:
```

Then delete the rule **number**
`iptables -t nat -D PREROUTING 2`

- **iptables**: How to do logging of iptbales

  create a new rule chain that logs and drops in sequence:

```
# Create a new chain called LOGGING
iptables -N LOGGING
#All the remaining incoming packets will jump to the LOGGING chain
iptables -A INPUT -j LOGGING
#Log the incoming packets to syslog (/var/log/messages)
iptables -A LOGGING -m limit --limit 3/min -j LOG --log-prefix "iptables dr
#Finally, drop all the packets that came to the LOGGING chain
iptables -A LOGGING -j DROP
```

  Log All Dropped Outgoing Packets

```
iptables -A LOGGING -m limit --limit 3/min -j LOG --log-prefix "iptables dr
iptables -A LOGGING -j DROP
```

- **iptables: How to build DDoS Rule in iptables**

```
# Reject spoofed packets
iptables -A INPUT -s 10.0.0.0/8 -j DROP
iptables -A INPUT -s 169.254.0.0/16 -j DROP
iptables -A INPUT -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i eth0 -s 127.0.0.0/8 -j DROP

iptables -A INPUT -s 224.0.0.0/4 -j DROP
iptables -A INPUT -d 224.0.0.0/4 -j DROP
iptables -A INPUT -s 240.0.0.0/5 -j DROP
iptables -A INPUT -d 240.0.0.0/5 -j DROP
iptables -A INPUT -s 0.0.0.0/8 -j DROP
iptables -A INPUT -d 0.0.0.0/8 -j DROP
iptables -A INPUT -d 239.255.255.0/24 -j DROP
iptables -A INPUT -d 255.255.255.255 -j DROP

# Stop smurf attacks
iptables -A INPUT -p icmp -m icmp --icmp-type address-mask-request -j DROP
iptables -A INPUT -p icmp -m icmp --icmp-type timestamp-request -j DROP
iptables -A INPUT -p icmp -m icmp -j DROP

# Drop all invalid packets
iptables -A INPUT -m state --state INVALID -j DROP
iptables -A FORWARD -m state --state INVALID -j DROP
iptables -A OUTPUT -m state --state INVALID -j DROP
```

```
# Drop excessive RST packets to avoid smurf attacks
iptables -A INPUT -p tcp -m tcp --tcp-flags RST RST -m limit --limit 2/seco
```

- **iptables How to block portscans**

```
# Anyone who tried to portscan us is locked out for an entire day.
iptables -A INPUT   -m recent --name portscan --rcheck --seconds 86400 -j D
iptables -A FORWARD -m recent --name portscan --rcheck --seconds 86400 -j D

# Once the day has passed, remove them from the portscan list
iptables -A INPUT   -m recent --name portscan --remove
iptables -A FORWARD -m recent --name portscan --remove

# These rules add scanners to the portscan list, and log the attempt.
iptables -A INPUT   -p tcp -m tcp --dport 139 -m recent --name portscan --s
iptables -A INPUT   -p tcp -m tcp --dport 139 -m recent --name portscan --s

iptables -A FORWARD -p tcp -m tcp --dport 139 -m recent --name portscan --s
iptables -A FORWARD -p tcp -m tcp --dport 139 -m recent --name portscan --s
```

if i Missed out any rules, post a comment, I will add in the List

Thanku for reading !!! Give a Share for Support

2.5k Shares Share Tweet

Asking for donation sound bad to me, so i'm raising fund from **The Modern Cryptography** sample chapters here then decide.

Alternatively to support you can buy My all four Cryptography book Just $10.99

Four Cryptography Book

~~$85.96~~ SUGGESTED PRICE

$10.99 BUNDLE PRICE

YOU PAY $10.99

AUTHOR EARNS $8.79

YOU PAY (US$) $10.99

EU customers: Price excludes VAT. VAT is added during checkout.

- The Modern Cryptography Book.
- Go lang Cryptography for developers
- Python Cryptography
- Cryptography for JavaScript Developer

Linux Related Topics

- Auto-mounting an NFS share using autofs
- Add Ipv6 Support to ECS

- IPtables Must Known Stuff
- How to Monitor All executed Commands
- Top ps command for Monitor/Troubleshoot
- Linux Mount&Unmout revisit
- Detecting Linux distribution name
- Exploring Alibaba cloud Free Trial

Ansible Related Topics

- Manage System users (sudo/Non sudo)
- Ansible Variables Explained in Roles
- Pass sudo/ssh password without prompt
- Ansible Windows Commands

Kubernetes Related Topics

- kubernetes install on using ansible
- kube install on in centos7/ubuntu7
- kubernetes Dashbaord Setup
- Pod,Cluster,Deploy,ReplicaSet Light Dive
- kubernetes secure nginx deployment
- kubernetes Port, Targetport and NodePort
- kubernetes Namespace
- kubenetes Auth,Authorization,Admission
- kubernetes Role-Based Access Control
- Kubernetes Privilege Escalation Vulnerability
- Prometheus Dashboard Access
- Kubernetes mysql

- Kubernetes Jenkins installation
- Kubernetes mariadb installation
- Kubernetes wordpress installation
- Kubernetes drupal installation
- Kubernetes traefik installation
- Kubernetes Ingress traefik
- kubernetes service external ip pending ?
- Service Mesh With Istio
- kubernetes Java client example
- Docker Private repo with SSL and AUTH

Openstack Articles

- Creating Cloud Images
- Image

- Openstack Client Installation
- FreeNas installation in Openstack
- virt-install Error Guest name is already in use

Applied Cryptography Topics

- Anatomy of GPG
- PKI in Nutshell
- OCSP in Nutshell
- TLSv1.3 in Nutshell
- Secure Nginx Configuration
- Java Keytool/Keystore in Nutshell
- Twenty Steps of SSH hardening
- Apache TLS Configuration

Web Crypto API Topics

- ECDH generateKey/Encrypt/Decrypt

python Cryptography Topics

- RSA Cryptography

PHP Cryptography Topics

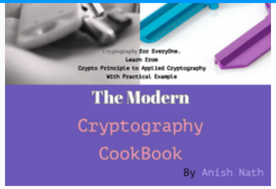
- openssl_get_cipher_methods
- openssl_get_md_methods
- openssl_digest
- openssl_cipher_iv_length
- openssl_encrypt/decrypt
- openssl_pkey_new/(rsa/dsa/ec)
- openssl_get_curve_names
- openssl_pbkdf2

Topics

- Cryptography
- Openstack
- Ansible
- Kubernetes
- Security
- Linux
- Go Lang Cryptography
- Python Cryptography

For Coffee/ Beer/ Amazon Bill and further

development of the project Support by Purchasing, The Modern Cryptography CookBook for Just $9 Coupon Price



The Modern Cryptography CookBook

Cryptography is for EveryOne. Learn...

Anish Nath

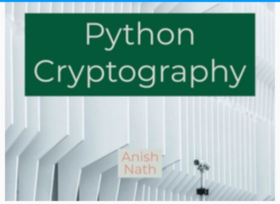
Buy Now

Minimum price: $14.99

Suggested price: $24.99

Kubernetes for DevOps

Cryptography for Python Developers



Python Cryptography

Python Cryptography

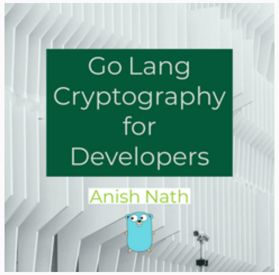
Anish Nath

Buy Now

Minimum price: $14.99
Suggested price: $24.99

Cryptography for JavaScript Developers

Go lang ryptography for Developers



GoLang Cryptography for Developer

Anish Nath

Buy Now

Minimum price: $13.99
Suggested price: $19.99

0 Comments

Sort by Oldest ⇕

Add a comment...

Facebook Comments Plugin