# How to build, configure and maintain a FreeBSD firewall with PF, BIND (DNS) and Squid (Proxy Server).

Twelve years ago, I couldn't afford the types of firewalls that were available at the time. They were big and expensive. Today, the cost has dropped dramatically and they are integrated into pretty much every cable modem and wireless access point. So the question comes, why would a person want to spend any of their time building a firewall when they could procure one for less than $100? The simple answer to this question is the search for knowledge. If you are of the mindset that you like understanding how the underlying technology works and enjoy periodically analyzing network traffic to further your knowledge base, then this article is for you. With a functional firewall, you can easily run tcpdump captures of network traffic and run saved captures through WireShark.

The platform of choice for this article will be FreeBSD 8.1 which you can obtain from http://www.freebsd.org. I chose this platform because I really enjoying the thoroughness of the documentation:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/.

Their documentation covers every aspect of what you need to achieve and is updated religiously. FreeBSD also comes with the binaries for BIND (link) and PF (link), so you will spend less time getting software and more time performing what's important to you (whether that is doing something else or performing network analysis on this system). These two applications will do

the bulk of the heavy lifting for the system – associating common names (URLs) with IP addresses and deciding what packets are allowed and which are to be blocked.

The hardware is an old Dell Precision with two physical cores. It is not 64 bit and has 2 GB of RAM. I've put two old IDE hard drives into the system, one is an IBM Deskstar (Primary Master) that is 9 GB and the other is a no name brand (Secondary Master) that has 30 GB of space. You don't have to have two disks, but I chose to follow this pattern because I want to split the IO load going to the disks. The larger hard drive will host the data for Squid, while the smaller one will serve the OS and swap space. I would recommend at least a 1GHz processor with at least 512MB of memory and at least a 30 GB hard drive. This system will work with much less, but it will run very slow. You will be much happier with these minimum requirements.

Lastly before we begin, the site that I have deployed at has three interfaces on the firewall. One attaches to a cable modem that routes traffic outbound. This cable modem has a software firewall installed and configured with the default settings from my ISP. The second interface attaches to a wireless access point to allow employee laptops to connect. This is secured with WPA2, MAC address control and has a non-broadcasting SSID. The third interface connects to servers and workstations. For the normal home/small office setup, you really only need two network interfaces. One for the outside interface that connects to your ISP cable modem and one that connects to a small switch that will connect all of your computer/networking equipment.

## OS Install

Before installing the OS, thoroughly read the following section of the FreeBSD handbook: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/install.html.

Partition the hard drive as needed for your site. I've set up the following partitions:

| Partition | Recommended Size |
|---|---|
| / | 1 GB |
| swap | 2 GB |
| /usr | 3 GB |
| /tmp | 512 MB |
| /var | remaining space hard drive 1 |
| /opt | all space on hard drive 2 |

When you get to the point of choosing Distributions, select Minimal and then proceed. I chose this because I want the least amount of software and services running to maximize the limited amount of memory on this system for the components I've selected (PF, BIND and Squid).

## OS Install Post Configuration

One of the most critical files on the system is /etc/rc.conf. This file is used to store all Multi-User mode configuration settings (this sounds like a Dr. Suess tongue twister, but means that this one file contains information about all installed applications and their startup settings, plus the network interface settings). Change this file to match your site requirements. Refer to the Appendix and the /etc/rc.conf sample.

I recommend running a dmesg or scan through /var/log/messages to determine the device names for the NICs (run "dmges | less" to see the device names). Change the file rc.conf to match your sites needs. Restart the network with the command, "/etc/rc.d/netif restart" after changing the references to the NICs inside of /etc/rc.conf. Again, refer to the Appendix and the sample code, this is very straight forward.

## BIND DNS server install

FreeBSD currently comes with BIND9 DNS server software by default.

## Squid proxy server install

On FreeBSD there are many ways to download and install software. Probably the easiest if you are connected to the internet is by using the pkg_add command with the flags "-vr". Issue the following command as root, "pkg_add –vr squid". You can also use the portsnap collection and compile quickly the Squid software. Read the handbook to understand this capability.

## PF firewall install

FreeBSD currently comes with the OpenBSD packet filtering software by default.

## Configure BIND

Activating the software was performed in the file /etc/rc.conf. The configuration file is /var/named/etc/namedb/named.conf. Refer to the Appendix for samples. Modify your configuration files to match your site requirements. Refer back to the FreeBSD handbook as needed for understanding how the files work together. Again, the documentation is excellent and doesn't need to be duplicated here.

## Test BIND from remote client

From the command line, use nslookup to test the remote DNS server.

```
$ nslookup
> server 192.168.5.1              ← tell nslookup to use 192.168.5.1 for DNS queries
> darthvader.fortress.lan         ← checking the name of my new firewall
Server: darthvader.fortress.lan
Address: 192.168.5.1

Name: darthvader.fortress.lan
Address: 192.168.5.1              ← this is the response I want to see, his own IP.

> www.cnn.com                     ← test external address
Server: darthvader.fortress.lan
Address: 192.168.5.1

Non-authoritative answer:
Name: www.cnn.com
Addresses: 157.166.224.26         ← this is the response I want to see.
           157.166.226.25
           157.166.226.26
           157.166.255.18
           157.166.255.19
```

## Configure Squid

Activating the software was performed in the file /etc/rc.conf. I used this link to install and startup WebMin: http://www.cyberciti.biz/faq/freebsd-installing-webmin/, I used the WebMin admin page to then configure Squid. You will need to read the documentation on: http://www.squid-cache.org/ to understand what's going on for squid. Basically, I have setup client authentication that forces the user to supply a username/password for web connections. Review my sample configuration file to understand what's going on and then compare against the documentation from squid-cache.org. Squid by default is not very complicated to configure and get running. However, if you deviate from the default configuration, it can become very complicated quickly and potentially stop working. When debugging, review the log files for the OS and for Squid, /var/log/messages and /opt/squid/logs/debug.log respectively.

## Test Squid from remote client

Configure your browser to use proxy port 3128 on the internal NIC (either TCB or DMZ). i.e. I used 192.168.5.1. Go to a sample URL, i.e. http://www.cnn.com. If your browser doesn't connect, make sure the service is running. Execute "/usr/local/etc/rc.d/squid restart". If this still doesn't work, it is because the firewall hasn't been configured and told to perform NAT'ing. Configure the firewall and then retest this component.

## Configure PF

Activating the software was performed in the file /etc/rc.conf. Refer to the sample file /usr/local/etc/pf.conf. Modify your site as needed to match the requirements.

## Test PF from remote client

Restart the PF binary with the command, "/etc/rc.d/pf restart". Open a command prompt or terminal and ping the firewall's IP address. Open a web browser (configured to use the proxy port) and connect to an external site.

## Positive and negative test cases

Positive: Ping an external site.
SSH to ip address of firewall.
Configure the proxy port in your web browser and try to connect to an external site.
Perform a DNS query to the firewall's DNS server.

Negative: Unconfigure the proxy port in your web browser and try to connect to an external site.
SSH to an external site.
Perform a DNS query to an external DNS server.

## Maintenance of packages and Operating System

Review the sample root crontab in the Appendix. I strongly recommend that you follow a similar pattern using the command freebsd-update. This will ensure your software packages are always up-to-date.

## Conclusion

At the end of this journey you should have a solid working firewall that blocks all traffic coming inbound and only allows outbound traffic through the proxy server. I've put all of my relevant configuration files and scripts into the Appendix. Along this journey, you should have read the documentation in the FreeBSD handbook (at least read through the installation procedures), learned BIND and Squid. The last step after the system is working is to gain an understanding of tcpdump. Read this site first, http://danielmiessler.com/study/tcpdump/. Then start analyzing small dumps of traffic from the system through a tool like Wireshark. I like to start off broad and then narrow my searches. I will SSH into the firewall and run something like:

```
tcpdump –i sk2 –c 10000 –w /tmp/quick.dmp –s 1524 ‘ip and not tcp port 64261’
```

After tcpdump has written 10,000 packets, I will secure copy the file over to a workstation with Wireshark installed and analyze the contents. You should learn all sorts of useful information from this exercise. From network broadcasts to mis-configured devices/services, you will learn what is happening on the network segment and hopefully be able to make better business decisions that shape your sites policies and guidelines. By analyzing the DNS queries performed, you can gain an understanding of where people and systems are trying to connect to. From this, you can choose to perform DNS blackholing to prevent unwanted traffic from being able to resolve the destination IP. Finally, with the logs from the squid proxy server, you can analyze these and make the decision to block sites if you choose. You can setup an access control list (ACL) in Squid that reads from a file and blocks those destinations. Don’t forget if you do this that every connection that goes through Squid will perform this lookup. So if you have many addresses in this file, your server’s performance will suffer.

## Contact and disclaimer information

Securityhardening.com provides ideas, concepts and patterns regarding computer security. This site and its content are intended as a way to share these thoughts with the security community. Any questions, comments, or complaints can be directed to the contact address on the support page. Anyone considering implementing any recommendation published on Securityhardening.com (in any form) should read the Disclaimer page.

-- Secure the system ... live life!

## Appendix

/etc/rc.conf sample:

```
#
hostname=”darthvader.fortress.lan”
#
ifconfig_sk0=”inet 192.168.5.1  netmask 255.255.255.0”
ifconfig_sk1=”inet 172.31.253.1 netmask 255.255.255.0”
ifconfig_sk2=”inet 10.101.101.1 netmask 255.255.255.0”
defaultrouter=”10.101.101.2”
#
gateway_enable=”YES”
inetd_enable=”NO”
sshd_enable=”YES”
#
pf_enable=”YES”
pf_rules=”/usr/local/etc/pf.conf”
pf_flags=””
pflog_enable=”YES”
pflog_logfile=”/var/log/pflog”
pflog_flags=””
#
ntpd_enable=”YES”
ntpd_sync_on_start=”NO”
#
update_motd=”NO”
#
dhcpd_enable=”YES”
#
named_enable=”YES”
#
squid_enable=”YES”
#
apache22_enable=”YES”
webmin_enable=”YES”
```

---

/var/named/etc/namedb/named.conf sample:

```
// Fortress.lan DNS config.
//
acl "friendlies" {
        127.0.0.1/32;
        192.168.5.0/24;
};

acl "dmz" {
        172.31.253.0/24;
};

options {
   directory       "/etc/namedb";
   pid-file        "/var/run/named/pid";
   dump-file       "/var/dump/named_dump.db";
```

```
   statistics-file "/var/stats/named.stats";
   //
   datasize 32768k;
   version "Nunya Business Sucka";
   transfer-format many-answers;
   recursion yes;
   allow-query { "friendlies"; "dmz"; };
   allow-transfer { "none"; };
   allow-update { "none"; };
   stacksize 32768k;
   //
   listen-on       { 127.0.0.1; 192.168.5.1; 172.31.253.1; };
   //
   disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
   disable-empty-zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
   disable-empty-zone
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";

};


logging {
        channel my_debug {
                    file "/var/log/named_debug.log" versions 9 size 5m;
                    severity debug 5;
                    print-time yes;
                    print-severity yes;
                    print-category yes;
        };
        channel security_info {
                    file "/var/log/named_auth.log" versions 9 size 5m;
                    severity notice;
                    print-time yes;
                    print-severity yes;
                    print-category yes;
        };
        channel info {
                    file "/var/log/named_info.log" versions 9 size 5m;
                    severity info;
                    print-time yes;
                    print-severity yes;
                    print-category yes;
        };
        category update { my_debug; };
        category security { security_info; };
        category queries { info; };
};

// RFC 1912
zone "localhost"        { type master; file "master/localhost-forward.db"; };
zone "127.in-addr.arpa" { type master; file "master/localhost-reverse.db"; };
zone "255.in-addr.arpa" { type master; file "master/empty.db"; };

// RFC 1912-style zone for IPv6 localhost address
zone "0.ip6.arpa"       { type master; file "master/localhost-reverse.db"; };

// "This" Network (RFCs 1912 and 3330)
zone "0.in-addr.arpa"           { type master; file "master/empty.db"; };

// Private Use Networks (RFC 1918)
//zone "10.in-addr.arpa"                { type master; file "master/empty.db"; };
//zone "16.172.in-addr.arpa"      { type master; file "master/empty.db"; };
```

```
zone "16.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "17.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "18.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "19.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "20.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "21.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "22.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "23.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "24.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "25.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "26.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "27.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "28.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "29.172.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "30.172.in-addr.arpa"      { type master; file "master/empty.db"; };
//zone "168.192.in-addr.arpa"     { type master; file "master/empty.db"; };

// Link-local/APIPA (RFCs 3330 and 3927)
zone "254.169.in-addr.arpa"     { type master; file "master/empty.db"; };

// TEST-NET for Documentation (RFC 3330)
zone "2.0.192.in-addr.arpa"     { type master; file "master/empty.db"; };

// Router Benchmark Testing (RFC 3330)
zone "18.198.in-addr.arpa"      { type master; file "master/empty.db"; };
zone "19.198.in-addr.arpa"      { type master; file "master/empty.db"; };

// IANA Reserved - Old Class E Space
zone "240.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "241.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "242.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "243.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "244.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "245.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "246.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "247.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "248.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "249.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "250.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "251.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "252.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "253.in-addr.arpa"         { type master; file "master/empty.db"; };
zone "254.in-addr.arpa"         { type master; file "master/empty.db"; };

// IPv6 Unassigned Addresses (RFC 4291)
zone "1.ip6.arpa"               { type master; file "master/empty.db"; };
zone "3.ip6.arpa"               { type master; file "master/empty.db"; };
zone "4.ip6.arpa"               { type master; file "master/empty.db"; };
zone "5.ip6.arpa"               { type master; file "master/empty.db"; };
zone "6.ip6.arpa"               { type master; file "master/empty.db"; };
zone "7.ip6.arpa"               { type master; file "master/empty.db"; };
zone "8.ip6.arpa"               { type master; file "master/empty.db"; };
zone "9.ip6.arpa"               { type master; file "master/empty.db"; };
zone "a.ip6.arpa"               { type master; file "master/empty.db"; };
zone "b.ip6.arpa"               { type master; file "master/empty.db"; };
zone "c.ip6.arpa"               { type master; file "master/empty.db"; };
zone "d.ip6.arpa"               { type master; file "master/empty.db"; };
zone "e.ip6.arpa"               { type master; file "master/empty.db"; };
zone "0.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "1.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "2.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "3.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "4.f.ip6.arpa"             { type master; file "master/empty.db"; };
```

```
zone "5.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "6.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "7.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "8.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "9.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "a.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "b.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "0.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "1.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "2.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "3.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "4.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "5.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "6.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "7.e.f.ip6.arpa"           { type master; file "master/empty.db"; };

// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa"             { type master; file "master/empty.db"; };
zone "d.f.ip6.arpa"             { type master; file "master/empty.db"; };

// IPv6 Link Local (RFC 4291)
zone "8.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "9.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "a.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "b.e.f.ip6.arpa"           { type master; file "master/empty.db"; };

// IPv6 Deprecated Site-Local Addresses (RFC 3879)
zone "c.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "d.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "e.e.f.ip6.arpa"           { type master; file "master/empty.db"; };
zone "f.e.f.ip6.arpa"           { type master; file "master/empty.db"; };

// IP6.INT is Deprecated (RFC 4159)
zone "ip6.int"                  { type master; file "master/empty.db"; };

// Master zones
zone "fortress.lan" {
        type master;
        file "master/fortress.lan";
        allow-query { "friendlies"; };
};

zone "99.99.10.in-addr.arpa" {
        type master;
        file "master/99.99.99.in-addr.arpa";
        allow-query { "friendlies"; };
};

zone "101.101.10.in-addr.arpa" {
        type master;
        file "master/101.101.10.in-addr.arpa";
        allow-query { "friendlies"; };
};

zone "5.168.192.in-addr.arpa" {
        type master;
        file "master/5.168.192.in-addr.arpa";
        allow-query { "friendlies"; };
};

zone "253.31.172.in-addr.arpa" {
        type master;
        file "master/253.31.172.in-addr.arpa";
```

```
        allow-query { "friendlies"; };
};

zone "." {
        type hint;
        file "named.root";
};
```

---

/var/named/etc/namedb/named.root sample:

```
;
; $FreeBSD: src/etc/namedb/named.root,v 1.12.18.2.4.1 2009/04/15 03:14:26 kensmith Exp
$
;

;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:    Feb 04, 2008
;       related version of root zone:   2008020400
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:BA3E::2:30
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     192.228.79.201
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
F.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:2f::f
;
```

```
; formerly NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
H.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:1::803f:235
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
;
; operated by VeriSign, Inc.
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     192.58.128.30
J.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:C27::2:30
;
; operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
K.ROOT-SERVERS.NET.      3600000      AAAA  2001:7fd::1
;
; operated by ICANN
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     199.7.83.42
;
; operated by WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.      3600000      AAAA  2001:dc3::35
; End of File
```

---

/var/named/etc/namedb/master/101.101.10.in-addr.arpa sample:

```
$TTL   86400

@      IN SOA  fortress.lan.      darthvader.fortress.lan.        (
              2003102600   ; Serial
              10800        ; Refresh (3 hours)
              3600         ; Retry (1 hour)
              604800       ; Expire (1 week)
              86400  )     ; Minimum (1 day)

              IN NS  darthvader.fortress.lan.
1             IN PTR  shumai.fortress.lan.
2             IN PTR  pooja.fortress.lan.
```

---

/var/named/etc/namedb/master/253.31.172.in-addr.arpa sample:

```
$TTL   86400
```

```
@      IN SOA  fortress.lan.      darthvader.fortress.lan.        (
              2003102600   ; Serial
              10800        ; Refresh (3 hours)
              3600         ; Retry (1 hour)
              604800       ; Expire (1 week)
              86400  )     ; Minimum (1 day)

              IN NS  darthvader.fortress.lan.
1             IN PTR  seitaria.fortress.lan.
2             IN PTR  darthmaul.fortress.lan.
```

---

/var/named/etc/namedb/master/5.168.192.in-addr.arpa sample:

```
$TTL   86400

@      IN SOA  fortress.lan.      darthvader.fortress.lan.        (
              2003102600   ; Serial
              10800        ; Refresh (3 hours)
              3600         ; Retry (1 hour)
              604800 ; Expire (1 week)
              86400  )     ; Minimum (1 day)

       IN NS  darthvader.fortress.lan.

1      IN PTR darthvader.fortress.lan.
2      IN PTR palpatine.fortress.lan.
3      IN PTR landocalrissian.fortress.lan.
4      IN PTR countdooku.fortress.lan.
5      IN PTR bobafett.fortress.lan.
```

---

/var/named/etc/namedb/master/fortress.lan sample:

```
$TTL   86400

@      IN SOA  fortress.lan.      darthvader.fortress.lan. (
              2003102600   ; Serial
              10800        ; Refresh (3 hours)
              3600         ; Retry (1 hour)
              604800       ; Expire (1 week)
              86400  )     ; Minimum (1 day)

              IN     NS     darthvader.fortress.lan.

localhost     IN     A      127.0.0.1

; DMZ & Firewall
shumai        IN     A      10.101.101.1
pooja         IN     A      10.101.101.2

; DMZ
seitaria      IN     A      172.31.253.1
darthmaul     IN     A      172.31.253.2

; Internal Fortress
darthvader    IN     A      192.168.5.1
palpatine     IN     A      192.168.5.2
landocalrissian     IN     A      192.168.5.3
countdooku    IN     A      192.168.5.4
bobafett      IN     A      192.168.5.5
```

/usr/local/etc/squid/squid.conf sample:

```
auth_param basic program /usr/local/libexec/squid/ncsa_auth \
/usr/local/etc/squid/squid.passwd
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
#############################################################################
acl windowsupdate dstdomain windowsupdate.microsoft.com
acl windowsupdate dstdomain .update.microsoft.com
acl windowsupdate dstdomain download.windowsupdate.com
acl windowsupdate dstdomain redir.metaservices.microsoft.com
acl windowsupdate dstdomain images.metaservices.microsoft.com
acl windowsupdate dstdomain c.microsoft.com
acl windowsupdate dstdomain www.download.windowsupdate.com
acl windowsupdate dstdomain wustat.windows.com
acl windowsupdate dstdomain crl.microsoft.com
acl windowsupdate dstdomain sls.microsoft.com
acl windowsupdate dstdomain productactivation.one.microsoft.com
acl windowsupdate dstdomain ntservicepack.microsoft.com
#############################################################################
acl CONNECT method CONNECT
acl wuCONNECT dstdomain www.update.microsoft.com
acl wuCONNECT dstdomain sls.microsoft.com
acl employees ident employee_username1 employee_username2 employee_username3
acl managers  ident manager_username1 manager_username2
acl pam proxy_auth REQUIRED
#############################################################################
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl localnet src 172.31.253.0/24 # RFC1918 possible internal network
acl localnet src 192.168.5.0/24 # RFC1918 possible internal network
acl SSL_ports port 443 # RFC1918 possible internal network
acl Safe_ports port 80
acl Safe_ports port 21 # http
acl Safe_ports port 443 # ftp
acl Safe_ports port 70 # https
acl Safe_ports port 210 # gopher
acl Safe_ports port 1025-65535 # wais
acl Safe_ports port 280 # unregistered ports
acl Safe_ports port 488 # http-mgmt
acl Safe_ports port 591 # gss-http
acl Safe_ports port 777 # filemaker
acl CONNECT method CONNECT # multiling http
acl shoutcast rep_header X-HTTP09-First-Line ^ICY.[0-9]
acl ncsa_users proxy_auth REQUIRED
acl business_hours_week    time M T W H F 16:00-19:00
acl business_hours_weekend time       A S  8:00-19:00
acl Employees.Good.Sites dstdomain "/usr/local/etc/squid/employee.allowed-sites.squid"
#############################################################################
http_access allow CONNECT wuCONNECT localnet
http_access allow windowsupdate localnet
http_access allow ncsa_users
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
```

```
http_access deny to_localhost
http_access allow localnet
http_access deny all
############################################################################
icp_access allow localnet
icp_access deny all
############################################################################
ident_lookup_access allow all
http_port 192.168.5.1:3128
http_port 172.31.253.1:3128
############################################################################
hierarchy_stoplist cgi-bin ?
cache_mem 256 MB
memory_replacement_policy heap LFUDA
cache_replacement_policy heap LFUDA
cache_dir ufs /opt/squid/cache 10000 16 256
maximum_object_size 8192 KB
access_log /opt/squid/logs/access.log
cache_log /opt/squid/logs/debug.log
cache_store_log /opt/squid/logs/storage.log
############################################################################
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
############################################################################
quick_abort_min -1
range_offset_limit -1
half_closed_clients off
store_avg_object_size 14 KB
cache_swap_high 100%
cache_swap_low 80%
acl apache rep_header Server ^Apache
upgrade_http0.9 deny shoutcast
broken_vary_encoding allow apache
cache_effective_user squid
cache_mgr webmistress@fortress.lan
icp_port 0
dns_testnames www.cnn.com
dns_nameservers 127.0.0.1
fqdncache_size 8192
coredump_dir /var/tmp
cache_swap_log /opt/squid/logs/cache.log
cache_effective_group squid
visible_hostname darthvader
unique_hostname darthvader.fortress.lan
############################################################################
```

---

/usr/local/etc/pf.conf sample:

```
##### -----   MACROS ## Definition  ----- #####
#
#
tcb_if="sk0" ## Trusted Computing Base
dmz_if="sk1" ## DMZ - Wireless network
ext_if="sk2" ## External Network - to Screening Router
Loop="lo0"   ## Loopback address.
#
TCP_Flags = "S/SAFR modulate state"
#
TCP_OutBound = "{ 3128 }"
```

```
UDP_OutBound = "{ }"
#
TCB_Networks = "{ 192.168.5.0/24 }"
DMZ_Networks = "{ 172.31.253.0/24 }"


##### -----   OPTIONS   ----- #####
#
#
set limit { states 10000, frags 5000 }
set timeout { interval 30, frag 10 }
set timeout { tcp.first 120, tcp.opening 30, tcp.established 86400 }
set timeout { tcp.closing 900, tcp.finwait 45, tcp.closed 90 }
set timeout { udp.first 60, udp.single 30, udp.multiple 60 }
set timeout { icmp.first 20, icmp.error 10 }
set timeout { other.first 60, other.single 30, other.multiple 60 }
set fingerprints "/usr/local/etc/pf.os"
set ruleset-optimization basic
set optimization conservative
set block-policy return
set require-order yes
set state-policy if-bound
set loginterface $tcb_if
set skip on $Loop
set debug urgent


##### -----   NORMALIZATION   ----- #####
#
#
scrub  in on $ext_if all fragment reassemble
scrub out on $ext_if all fragment reassemble random-id no-df


##### -----   TRANSLATION   ----- #####
#
#
nat on $ext_if from 192.168.5.0/24 to any -> ($ext_if)
nat on $ext_if from 172.31.253.0/24 to any -> ($ext_if)


##### -----   PACKET FILTERING   ----- #####
#
#
# Defaults
block in log all
pass out all

##### Loopback specific rules:
pass in  quick on lo0 all
pass out quick on lo0 all
antispoof log for lo0


#Allow Restrictive UDP From TCB to firewall DNS
pass in quick on { $tcb_if } proto udp from \
     $TCB_Networks to { $tcb_if } \
     port 53 keep state

#Allow Restrictive UDP From DMZ to firewall DNS
pass in quick on { $dmz_if } proto udp from \
     $DMZ_Networks to { $dmz_if } \
     port 53 keep state
```

```
#Allow Less Restrictive TCP From TCB to External
pass in quick on { $tcb_if } proto tcp from \
     $TCB_Networks to { $tcb_if }  \
     port $TCP_OutBound flags $TCP_Flags

#Allow Less Restrictive TCP From DMZ to External
pass in quick on { $dmz_if } proto tcp from \
     $DMZ_Networks to { $dmz_if }  \
     port $TCP_OutBound flags $TCP_Flags

# Management stuff – SSH on 64261.
pass in quick on { $tcb_if } proto tcp from \
     $TCB_Networks to { $tcb_if } \
     port { 10000, 11000, 64261 } flags $TCP_Flags

#Allow Less Restrictive ICMP From Int to External
pass in quick on { $tcb_if } proto icmp from \
     $TCB_Networks to any \
     icmp-type 8 keep state
```

Note: my /usr/local/etc/pf.os is empty.

---

/etc/ntp.conf sample:

```
restrict default nomodify notrap noquery

server 127.127.1.0
restrict 127.0.0.1

server 192.5.41.40
restrict 192.5.41.40 mask 255.255.255.255 nomodify notrap noquery

server 192.5.41.41
restrict 192.5.41.41 mask 255.255.255.255 nomodify notrap noquery

fudge   127.127.1.0 stratum 10

driftfile /var/lib/ntp/drift
broadcastdelay  0.008

keys            /etc/ntp/keys
#broadcastclient
```

---

/etc/sysctl.conf sample:

```
#
kern.ps_showallprocs=0
#
net.inet.tcp.blackhole=1
net.inet.udp.blackhole=1
#
net.inet.tcp.log_in_vain=1
net.inet.udp.log_in_vain=1
#
net.inet.ip.forwarding=1
net.inet.ip.ttl=32
net.inet.icmp.icmplim=200
net.inet.icmp.bmcastecho=0
```

```
net.inet.icmp.maskrepl=0
#
net.inet.tcp.sendspace=32768
net.inet.tcp.recvspace=65536
net.inet.tcp.recvbuf_auto=1
net.inet.tcp.recvbuf_inc=16384
net.inet.tcp.recvbuf_max=262144
net.inet.tcp.sendbuf_auto=1
net.inet.tcp.sendbuf_inc=8192
net.inet.tcp.sendbuf_max=262144
#
net.inet.tcp.syncookies=0
net.inet.tcp.slowstart_flightsize=2
net.inet.tcp.local_slowstart_flightsize=5
net.inet.tcp.newreno=1
net.inet.tcp.tso=1
#
net.inet.tcp.sack.enable=1
net.inet.tcp.always_keepalive=1
net.link.log_link_state_change=1
#
```

---

/etc/ssh/sshd_config sample:

```
Port 64261
Protocol 2
PermitRootLogin no
X11Forwarding no
PrintMotd yes
PrintLastLog yes
KeepAlive yes
PermitEmptyPasswords no
PasswordAuthentication yes
ReverseMappingCheck no
GatewayPorts no
AllowTcpForwarding yes
AllowGroups ssh
Banner /etc/issue
MaxStartups 4
```

---

/root/bin/backup.sh sample:

```
#!/usr/bin/env bash
#
# backup critical system files
#
PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/root/bin"
BACKUP_DIR="/var/backups"
[ ! -d $BACKUP_DIR ] && mkdir -p $BACKUP_DIR
DATE=`date +%Y%m%d`
BACKUP_DAILY="$BACKUP_DIR/$DATE"
[ ! -d $BACKUP_DAILY ] && mkdir -p $BACKUP_DAILY

##
cd $BACKUP_DAILY
[ -f $DATE.tgz ] && rm -f $DATE.tgz

find /etc -depth -type f | pax -w -x tar | gzip -9 > etc.tgz
find /var/named/etc -depth -type f | pax -w -x tar | gzip -9 > var.named.etc.tgz
```

```
find /usr/local/etc -depth -type f | pax -w -x tar | gzip -9 > usr.local.etc.tgz

cd $BACKUP_DIR
tar cf - $DATE | gzip -9 > $DATE.tgz

rm -rf $BACKUP_DAILY

exit 0
```

---

Root crontab sample (crontab –l –u root):

```
#
PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/root/bin"
#
#   Critical System Files
02  02  *  *  *  /root/bin/backup.sh 2>/dev/null
#
#   System Updates
03  03  *  *  *  /usr/sbin/freebsd-update cron
#
#   Ports Tree update
04  04  *  *  *  /usr/sbin/portsnap cron update
```

---