anongw / blog

Production

Hard to believe, but in this two years I deployed the anonymous gateway in two companies (*thank you PRISM!*). One exactly built as in this (semi) tutorial, the other one ran by a rack server, equipped with a dual quad core Xeon (16 threads, 16GB RAM), clusterized to another box for **high availability**. I was gone really crazy the first weeks adding rules on multiple layers: iptables, privoxy, squid, vpn, clusterized dhcp and dns, etc. The main problem has been to exclude address of web banking for the administration office: https, distributed geographically servers, and paranoid security system are really an headache. I solved installing FoxyProxy in the clients and customizing the rules.

40 privoxy and tor instances over a **40** round robin squid cache proxy peers, **2** TOR TransPort on different instances each user's MacAddress to torify the ports different from *80* tcp.

(One i2p tunnel each **10** privoxy instances)

20 persons behind the anongw.

I have had work (paid) for more then 1 year and... I really enjoyed a lot

Posted gio 06 ago 2015 21:41:37 UTC

Tags: production

dnscrypt

DNSCrypt is a tool for securing communications between a client (our AnonGW) and a DNS resolver. Description

dnscrypt-proxy provides local service which can be used directly as your local resolver or as a DNS forwarder, encrypting and authenticating requests using the DNSCrypt protocol and passing them to an upstream server, by default OpenDNS who run this on their resolvers.

The DNSCrypt protocol uses high-speed high-security elliptic-curve cryptography and is very similar to DNSCurve, but focuses on securing communications between a client and its first-level resolver.

While **NOT** providing end-to-end security or **anonymity**, it protects the local network, which is often the weakest point of the chain, against man-in-the-middle attacks. It also provides some confidentiality to DNS queries. In our POC can be a good compromise between anonymity and full resolution of any type of queries that actually the tor system can't resolve.

← ott 2016 →

d	l	m	m	g	v	s
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Recent Comments

Archives

Tags:

- POC
- TOR
- abstract
- apparmor
- dnscrypt
- firewall
- intro
- iptables
- privoxy
- production
- routing
- squid3

Download the current version (in May 2013 the version is **1.3.0**):

```
wget http://download.dnscrypt.org/dnscrypt-proxy/dnscrypt-proxy
```

and **gpg --verify** the signature.

Compile and install it:

```
AnonGW:~$ bunzip2 -cd dnscrypt-proxy-*.tar.bz2 | tar xvf -
AnonGW:~$ cd dnscrypt-proxy-* && ./configure && make -j2
AnonGW:~# make install
```

This is an example of the init script:

```
AnonGW:~# cat /etc/init.d/dnscrypt
```

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          dnscrypt
# Required-Start:    $network $remote_fs $syslog
# Required-Stop:     $network $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start dnscrypt
# Description:       Encrypt DNS queries.
### END INIT INFO
# TEST http://www.opendns.com/welcome
DAEMON="/usr/local/sbin/dnscrypt-proxy"
NAME="dnscrypt"
dnscrypt_start()
{
    /bin/echo "Starting dnscrypt"
    /sbin/ifconfig lo:2 127.0.0.2 up
    dnscrypt-proxy --user=bind  --local-address=127.0.0.2:40 --
}
dnscrypt_stop()
{
    echo "Stopping dnscrypt"
    start-stop-daemon --oknodo --stop --quiet --retry=0/3/KILL/
    /sbin/ip addr del 127.0.0.2/8 dev lo label lo:2
}
case "$1" in
    start)
   dnscrypt_start
   ;;
  stop)
   dnscrypt_stop
  ;;
  restart|force-reload)
   dnscrypt_stop
  dnscrypt_start
   ;;
    *)
   echo "Usage: /etc/init.d/$NAME {start|stop|restart|force-rel
   exit 1
   ;;
```

```
esac
exit 0
```

Start and check:

```
AnonGW:~# /etc/init.d/dnscrypt start && netstat -nlapo|grep dns
Starting dnscrypt
tcp        0      0 127.0.0.2:40            0.0.0.0:*
udp        0      0 0.0.0.0:58146           0.0.0.0:*
udp        0      0 127.0.0.2:40            0.0.0.0:*
unix  3      [ ]         STREAM     CONNECTED      219530   308
unix  3      [ ]         STREAM     CONNECTED      219529   308
unix  2      [ ]         DGRAM                     219528   308
```

That's all folks.

Posted lun 27 mag 2013 16:53:08 UTC

Tags: dnscrypt

iptables

Below a part of a script to manage firewalling and routing/natting rules.

This is only an example and can be a function to include in a main script to run on boot or to use to add or remove devices from the anonymizing rules without change the default gateway.

```
#!/bin/bash

firewall () {

/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -t mangle -F
/sbin/iptables -X
/sbin/iptables -t nat -X
/sbin/iptables -t mangle -X
/sbin/iptables -Z
#_____________POLICY_____________________
/sbin/iptables -P INPUT DROP
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/ip6tables -P INPUT DROP
/sbin/ip6tables -P FORWARD DROP
/sbin/ip6tables -P OUTPUT DROP
#__________CUSTOM_ CHAINS________________
/sbin/iptables -N SSH_CHECK
/sbin/iptables -N SSH_ATTACKED
/sbin/iptables -N SCAN_CHECK
/sbin/iptables -t nat -N BYPASS
    #_________________MANGLE________________
/sbin/iptables -t mangle -A POSTROUTING -o $virbr -p udp -m udp --dport 68 -j CHECKS
/sbin/iptables -t mangle -A OUTPUT -p udp --dport 53 -j TOS --set-tos Minimize-Delay
/sbin/iptables -t mangle -A OUTPUT -p udp --dport 40 -j TOS --set-tos Minimize-Delay
/sbin/iptables -t mangle -A FORWARD -p tcp -m multiport --dport 22,873 -j TOS --set-
```

```
#
#___________________OUTPUT_____________________
# Custom
#___________________FORWARD____________________
#
#____________________INPUT_____________________
/sbin/iptables -A INPUT  -i lo -j ACCEPT
/sbin/iptables -A INPUT -i $IntBr -j ACCEPT
/sbin/iptables -A INPUT -i $ExtBr -m multiport -p udp --dports $vpn_port,$i2p_port,$
/sbin/iptables -A INPUT -i $ExtBr -m multiport -p tcp --dports $i2p_port,$gnunet_por
/sbin/iptables -A INPUT -i $ExtBr -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -i $lxcbr -j ACCEPT
#
# SSH rules to comply with hosts.allow
/sbin/iptables -A INPUT -i $ExtBr -p tcp -m state --state NEW --dport 22 -j SSH_CHEC
/sbin/iptables -A INPUT -i $tun -p tcp -m state --state NEW --dport 22 -j SSH_CHECK
/sbin/iptables -A INPUT -i $virbr -p tcp -m state --state NEW --dport 22 -j SSH_CHEC
/sbin/iptables -A INPUT -i $lxcbr -p tcp -m state --state NEW --dport 22 -j SSH_CHEC
/sbin/iptables -A INPUT -i $tun -j ACCEPT
/sbin/iptables -A INPUT -i $virbr -j ACCEPT
#
/sbin/iptables -A SSH_CHECK -m recent --set --name SSH
/sbin/iptables -A SSH_CHECK -m recent --update --seconds 180 --hitcount 6 --name SSH
/sbin/iptables -A SSH_CHECK -j ACCEPT
/sbin/iptables -A SSH_ATTACKED -j LOG --log-prefix "iptables SSH attack: " --log-lev
/sbin/iptables -A SSH_ATTACKED -j DROP
#
/sbin/iptables -A INPUT -i $ExtBr -p tcp --dport 22 -j DROP
#
/sbin/iptables -A INPUT -i $ExtBr -p icmp -j DROP
/sbin/iptables -A INPUT -i $ExtBr -p tcp --syn -m limit --limit 1/s -j ACCEPT
/sbin/iptables -A INPUT -i $ExtBr -p tcp --syn -j DROP
/sbin/iptables -A INPUT -i $ExtBr -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --
/sbin/iptables -A INPUT -i $ExtBr -p tcp --tcp-flags SYN,ACK,FIN,RST RST -j DROP
#
# NAT rules
/sbin/iptables -t nat -A POSTROUTING -o $tun -j MASQUERADE
/sbin/iptables -t nat -A POSTROUTING -o $ExtBr -p tcp ! -d $Ext_Net -j SNAT --to $IP
/sbin/iptables -t nat -A POSTROUTING -o $ExtBr -p udp ! -d $Ext_Net -j SNAT --to $IP
/sbin/iptables -t nat -A POSTROUTING -o $ExtBr ! -d $Ext_Net -j SNAT --to $IP_Ext --
/sbin/iptables -t nat -A POSTROUTING -s $vir_net ! -d $vir_net -p tcp -o $IntBr -j M
/sbin/iptables -t nat -A POSTROUTING -s $vir_net ! -d $vir_net -p udp -o $IntBr -j M
/sbin/iptables -t nat -A POSTROUTING -s $vir_net ! -d $vir_net -o $IntBr -j MASQUERA
/sbin/iptables -t nat -A POSTROUTING -s $lxc_net -j MASQUERADE
/sbin/iptables -t nat -A POSTROUTING -s $VPN -o $IntBr -j MASQUERADE
####### squid.conf: rules complying to --> acl localnet src
/sbin/iptables -t mangle -A PREROUTING -p tcp --dport 3128 ! -s $VPN -j DROP
#_____ SQUID _____
/sbin/iptables -t nat -A PREROUTING -s $DefaultGW -p tcp --dport 80 -j ACCEPT
#_____ TAHOE LAFS _____
/sbin/iptables -t nat -A PREROUTING -p tcp --dport 3456 -j REDIRECT --to-port 80

################  BYPASS RULES ###################
/sbin/iptables -t nat -A BYPASS -d 192.168/16,172.16/16 -j ACCEPT # or any remote LA
/sbin/iptables -t nat -A BYPASS -d $safe_pub_addr -j ACCEPT
/sbin/iptables -t nat -A BYPASS -p udp --dport 1194 -d $ovpn_server1 -j ACCEPT
/sbin/iptables -t nat -A BYPASS -p udp --dport 1194 -d $ovpn_server2 -j ACCEPT
##########################################################

################ Load Mac Address filter ##################
for i in `/usr/bin/find /usr/local/etc/anon/ -maxdepth 1 -name *.fw -type f -print`;
```

```
deep_PC
deep_laptop
deep_tablet
deep_smartphone
}

firewall
```

Where **`/usr/local/etc/anon/`** is a directory created to contain one single files per device.

```
AnonGW:~# ls /usr/local/etc/anon/
PC.fw laptop.fw tablet.fw smartphone.fw
```

and

```
AnonGW:~# cat /usr/local/etc/anon/*.fw

#################_______PC_______######################
deep_PC () {
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_PC -i $
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_PC -i $
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_PC --dp
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_PC --dp
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_PC --dp
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_PC -i $IntBr
}

#################_______LAPTOP_____######################
deep_minikatik () {
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_laptop
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_laptop
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_laptop
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_laptop
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_laptop
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_laptop -i $Int
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_laptop -i $I
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_laptop -i $Int
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_laptop -i $I
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_laptop -i $Int
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_laptop -i $I
}

#################____TABLET____######################
deep_tablet () {
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_tablet
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_tablet
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_tablet
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_tablet
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source $HWaddr_tablet
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_tablet -i $Int
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_tablet -i $I
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_tablet -i $Int
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_tablet -i $I
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source $HWaddr_tablet -i $Int
```

```
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source $HWaddr_tablet -i $
}

################_____SMARTPHONE____####################
deep_smartphone () {
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source smartphone  -i
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source smartphone -i
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source smartphone --d
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source smartphone --d
/sbin/iptables -t nat -A PREROUTING -m tcp -p tcp -m mac --mac-source smartphone --d
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source smartphone -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source smartphone -i $IntB
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source smartphone -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source smartphone -i $IntB
/sbin/iptables -t nat -A PREROUTING -p tcp -m mac --mac-source smartphone -i $IntBr
/sbin/iptables -t nat -A PREROUTING ! -p tcp -m mac --mac-source smartphone -i $IntB
}
```

Posted gio 23 mag 2013 15:58:40 UTC

Tags: firewall iptables routing

tor

Below the *n* configuration files for each *n* TOR instance (these are minimal configurations and obviously it's possible to add relay configurations or hidden services).

1th

```
AnonGW:~# cat /etc/tor/torrc
```

```
User debian-tor
SocksPort 9050 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.1.1:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor
ControlPort 9051
CookieAuthentication 1
```

2nd

```
AnonGW:~# cat /etc/tor/torrc2
```

```
User debian-tor
ControlSocket /var/run/tor/control2
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control2.authcookie
SocksPort 9150 # what port to open for local application connections
```

```
SocksListenAddress 127.0.0.1 # accept connections only from localhost
TransPort $DefaultGW:9443
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.2:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor2
ControlPort 9151
```

3rd

```
AnonGW:~# cat /etc/tor/torrc3
```

```
User debian-tor
ControlSocket /var/run/tor/control3
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control3.authcookie
SocksPort 9250 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.3:53
TransPort 9040
TransPort $DefaultGW:9040
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor3
ControlPort 9251
```

4th

```
AnonGW:~# cat /etc/tor/torrc4
```

```
User debian-tor
ControlSocket /var/run/tor/control4
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control4.authcookie
SocksPort 9350 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
SocksListenAddress $DefaultGW:9105 # listen on this IP:port also
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.4:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor4
ControlPort 9351
```

5th

```
AnonGW:~# cat /etc/tor/torrc5
```

```
User debian-tor
ControlSocket /var/run/tor/control5
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control5.authcookie
SocksPort 9450 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.5:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor5
ControlPort 9451
```

6th

```
AnonGW:~# cat /etc/tor/torrc6
```

```
User debian-tor
ControlSocket /var/run/tor/control6
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control6.authcookie
SocksPort 9550 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.6:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor6
ControlPort 9551
```

7th

```
AnonGW:~# cat /etc/tor/torrc7
```

```
User debian-tor
ControlSocket /var/run/tor/control7
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control7.authcookie
SocksPort 9650 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.7:53
Log notice syslog
RunAsDaemon 1
```

```
DataDirectory /var/lib/tor7
ControlPort 9651
```

8th

```
AnonGW:~# cat /etc/tor/torrc8
```

```
User debian-tor
ControlSocket /var/run/tor/control8
ControlSocketsGroupWritable 1
CookieAuthentication 1
CookieAuthFileGroupReadable 1
CookieAuthFile /var/run/tor/control8.authcookie
SocksPort 9750 # what port to open for local application connections
SocksListenAddress 127.0.0.1 # accept connections only from localhost
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion,.i2p.xyz
DNSPort 127.0.0.8:53
Log notice syslog
RunAsDaemon 1
DataDirectory /var/lib/tor8
ControlPort 9751
```

Note:

To prepare the *N* TOR instances we have to add data directories for each demon and we have to change the TOR related Apparmor configuration file:

```
root@AnonGW:~# mkdir /var/lib/tor{2,3,4,5,6,7,8} && chown debian-tor:debian-tor /var
```

APPARMOR configuration

```
AnonGW:~# cat /etc/apparmor.d/usr.sbin.tor
```

```
/usr/sbin/tor {
  #include <abstractions/base>
  #include <abstractions/nameservice>
  network tcp,
  network udp,
  capability chown,
  capability dac_override,
  capability fowner,
  capability fsetid,
  capability setgid,
  capability setuid,
  /proc/sys/kernel/random/uuid r,
  /sys/devices/system/cpu/ r,
  /sys/devices/system/cpu/** r,
  /etc/tor/* r,
  /usr/share/tor/** r,
  owner /var/lib/tor/** rwk,
  owner /var/lib/tor2/** rwk,
  owner /var/lib/tor3/** rwk,
  owner /var/lib/tor4/** rwk,
```

```
  owner /var/lib/tor5/** rwk,
  owner /var/lib/tor6/** rwk,
  owner /var/lib/tor7/** rwk,
  owner /var/lib/tor8/** rwk,
  owner /var/log/tor/* w,
  /{,var/}run/tor/control w,
  /{,var/}run/tor/control2 w,
  /{,var/}run/tor/control3 w,
  /{,var/}run/tor/control4 w,
  /{,var/}run/tor/control5 w,
  /{,var/}run/tor/control6 w,
  /{,var/}run/tor/control7 w,
  /{,var/}run/tor/control8 w,
  /{,var/}run/tor/tor.pid w,
  /{,var/}run/tor/tor2.pid w,
  /{,var/}run/tor/tor3.pid w,
  /{,var/}run/tor/tor4.pid w,
  /{,var/}run/tor/tor5.pid w,
  /{,var/}run/tor/tor6.pid w,
  /{,var/}run/tor/tor7.pid w,
  /{,var/}run/tor/tor8.pid w,
  /{,var/}run/tor/control.authcookie w,
  /{,var/}run/tor/control.authcookie.tmp rw,
  /{,var/}run/tor/control2.authcookie w,
  /{,var/}run/tor/control3.authcookie w,
  /{,var/}run/tor/control4.authcookie w,
  /{,var/}run/tor/control5.authcookie w,
  /{,var/}run/tor/control6.authcookie w,
  /{,var/}run/tor/control7.authcookie w,
  /{,var/}run/tor/control8.authcookie w,
  /{,var/}run/tor/control2.authcookie.tmp rw,
  /{,var/}run/tor/control3.authcookie.tmp rw,
  /{,var/}run/tor/control4.authcookie.tmp rw,
  /{,var/}run/tor/control5.authcookie.tmp rw,
  /{,var/}run/tor/control6.authcookie.tmp rw,
  /{,var/}run/tor/control7.authcookie.tmp rw,
  /{,var/}run/tor/control8.authcookie.tmp rw,
  # Site-specific additions and overrides. See local/README for details.
  #include <local/usr.sbin.tor>
}
```

Those are simple (horrible) scripts to manage multiple demons: [FIX]

```
AnonGW:~# cat /usr/local/bin/RestartTor
```

```
#!/bin/bash
/usr/sbin/service tor stop ; /bin/sleep 6 ; /usr/sbin/service tor start
```

```
AnonGW:~# cat /etc/init.d/tor
```

```
#! /bin/bash
### BEGIN INIT INFO
# Provides:          tor
# Required-Start:    $local_fs $remote_fs $network $named $time
# Required-Stop:     $local_fs $remote_fs $network $named $time
# Should-Start:      $syslog
# Should-Stop:       $syslog
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      0 1 6
# Short-Description: Starts The Onion Router daemon processes
# Description:       Start The Onion Router, a TCP overlay
#                    network client that provides anonymous
#                    transport.
### END INIT INFO
. /lib/init/vars.sh
. /lib/lsb/init-functions
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/sbin/tor
NAME=tor
DESC="tor daemon"
TORPIDDIR=/var/run/tor
TORPID=$TORPIDDIR/tor.pid
DEFAULTSFILE=/etc/default/$NAME
WAITFORDAEMON=60
DEFAULT_ARGS="--defaults-torrc /usr/share/tor/tor-service-defaults-torrc"
VERIFY_ARGS="--verify-config $DEFAULT_ARGS"
ARGS=""
if [ "${VERBOSE:-}" != "yes" ]; then
    ARGS="$ARGS --hush"
fi
if [ -r /proc/sys/fs/file-max ]; then
    system_max=`cat /proc/sys/fs/file-max`
    if [ "$system_max" -gt "80000" ] ; then
            MAX_FILEDESCRIPTORS=32768
    elif [ "$system_max" -gt "40000" ] ; then
            MAX_FILEDESCRIPTORS=16384
    elif [ "$system_max" -gt "10000" ] ; then
            MAX_FILEDESCRIPTORS=8192
    else
            MAX_FILEDESCRIPTORS=1024
            cat << EOF
Warning: Your system has very few filedescriptors available in total.
Maybe you should try raising that by adding 'fs.file-max=100000' to your
/etc/sysctl.conf file.  Feel free to pick any number that you deem appropriate.
Then run 'sysctl -p'.  See /proc/sys/fs/file-max for the current value, and
file-nr in the same directory for how many of those are used at the moment.
EOF
    fi
else
    MAX_FILEDESCRIPTORS=8192
fi
NICE=""
test -x $DAEMON || exit 0
if [ -f $DEFAULTSFILE ] ; then
    . $DEFAULTSFILE
fi
wait_for_deaddaemon () {
    pid=$1
    sleep 1
    if test -n "$pid"
    then
            if kill -0 $pid 2>/dev/null
            then
                    cnt=0
                    while kill -0 $pid 2>/dev/null
                    do
                            cnt=`expr $cnt + 1`
                            if [ $cnt -gt $WAITFORDAEMON ]
                            then
```

```
                                log_action_end_msg 1 "still running"
                        fi
                        sleep 1
                        [ "`expr $cnt % 3`" != 2 ] || log_action_cont_msg ""
                done
        fi
    fi
    log_action_end_msg 0
}
check_torpiddir () {
    if test ! -d $TORPIDDIR; then
        mkdir -m 02750 "$TORPIDDIR"
        chown debian-tor:debian-tor "$TORPIDDIR"
    fi
    if test ! -x $TORPIDDIR; then
        log_action_end_msg 1 "cannot access $TORPIDDIR directory, are you root?"
        exit 1
    fi
}
check_config () {
    if ! $DAEMON $VERIFY_ARGS > /dev/null; then
        log_failure_msg "Checking if $NAME configuration is valid"
        $DAEMON --verify-config >&2
        exit 1
    fi
}
case "$1" in
  start)
    if [ "$RUN_DAEMON" != "yes" ]; then
        log_action_msg "Not starting $DESC (Disabled in $DEFAULTSFILE)."
        exit 0
    fi
    if [ -n "$MAX_FILEDESCRIPTORS" ]; then
        [ "${VERBOSE:-}" != "yes" ] || log_action_begin_msg "Raising maximum num

        [ "${VERBOSE:-}" != "yes" ] || log_action_begin_msg "Raising maximum num
        if ulimit -n "$MAX_FILEDESCRIPTORS" ; then
                [ "${VERBOSE:-}" != "yes" ] || log_action_end_msg 0
        else
                [ "${VERBOSE:-}" != "yes" ] || log_action_end_msg 1
        fi
    fi
    check_torpiddir
    check_config
    log_action_begin_msg "Starting $DESC"
    if start-stop-daemon --stop --signal 0 --quiet --pidfile $TORPID --exec $DAEMON;
        log_action_end_msg 0 "already running"
    else
        if start-stop-daemon --start --quiet \
                --pidfile $TORPID \
                $NICE \
                        --exec $DAEMON -- $DEFAULT_ARGS $ARGS
        /usr/sbin/tor -f /etc/tor/torrc2 --hush --pidfile /var/run/tor/tor2.pid &
            /usr/sbin/tor -f /etc/tor/torrc3 --quiet --pidfile /var/run/tor/tor3
            /usr/sbin/tor -f /etc/tor/torrc4 --quiet --pidfile /var/run/tor/tor4
            /usr/sbin/tor -f /etc/tor/torrc5 --quiet --pidfile /var/run/tor/tor5
            /usr/sbin/tor -f /etc/tor/torrc6 --quiet --pidfile /var/run/tor/tor6
            /usr/sbin/tor -f /etc/tor/torrc7 --quiet --pidfile /var/run/tor/tor7
            /usr/sbin/tor -f /etc/tor/torrc8 --quiet --pidfile /var/run/tor/tor8
        then
                log_action_end_msg 0
```

```
            else
                    log_action_end_msg 1
            fi
    fi
    ;;
  stop)
    log_action_begin_msg "Stopping $DESC"
    pid=`cat $TORPID 2>/dev/null` || true
    if test ! -f $TORPID -o -z "$pid"; then
            log_action_end_msg 0 "not running - there is no $TORPID"
            exit 0
    fi
    if start-stop-daemon --stop --signal INT --quiet --pidfile $TORPID --exec $DAEMO
            wait_for_deaddaemon $pid
    killall tor
    elif kill -0 $pid 2>/dev/null; then
            log_action_end_msg 1 "Is $pid not $NAME?  Is $DAEMON a different binary
    else
            log_action_end_msg 1 "$DAEMON died: process $pid not running; or permiss
    fi
    ;;
  reload|force-reload)
    check_config
    log_action_begin_msg "Reloading $DESC configuration"
    pid=`cat $TORPID 2>/dev/null` || true
    if test ! -f $TORPID -o -z "$pid"; then
            log_action_end_msg 1 "not running - there is no $TORPID"
            exit 0
    fi
    if start-stop-daemon --stop --signal 1 --quiet --pidfile $TORPID --exec $DAEMON
    then
            log_action_end_msg 0
    elif kill -0 $pid 2>/dev/null; then
            log_action_end_msg 1 "Is $pid not $NAME?  Is $DAEMON a different binary
    else
            log_action_end_msg 1 "$DAEMON died: process $pid not running; or permiss
    fi
    ;;
  restart)
    check_config
    $0 stop
    sleep 1
    $0 start
    ;;
  status)
    if test ! -r $(dirname $TORPID); then
            log_failure_msg "cannot read PID file $TORPID"
            exit 4
    fi
    pid=`cat $TORPID 2>/dev/null` || true
    if test ! -f $TORPID -o -z "$pid"; then
            log_failure_msg "$NAME is not running"
            exit 3
    fi
    if ps "$pid" >/dev/null 2>&1; then
            log_success_msg "$NAME is running"
            exit 0
    else
            log_failure_msg "$NAME is not running"
            exit 1
    fi
```

```
    ;;
  *)
```

Posted ven 17 mag 2013 10:40:47 UTC

Tags: TOR apparmor

privoxy

For clustering and managing *n* TOR instances we need *n* concurrent Privoxy processes:

1th

```
AnonGW:~# cat /etc/privoxy/config
```

```
user-manual /usr/share/doc/privoxy/user-manual
confdir /etc/privoxy
logdir /var/log/privoxy
actionsfile match-all.action # Actions that are applied to all sites and maybe overr
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
filterfile user.filter      # User customizations
logfile logfile
listen-address  127.0.0.1:8118
listen-address  $IntNIC:8443
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9050  .
forward-socks4a   /               127.0.0.1:9050  .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
socket-timeout 300
```

2nd

```
AnonGW:~# cat /etc/privoxy/config2
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy2
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8129
```

```
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9150 .
forward-socks4a   /               127.0.0.1:9150 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

3rd

```
AnonGW:~# cat /etc/privoxy/config3
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy3
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8230
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9250 .
forward-socks4a   /               127.0.0.1:9250 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

4th

```
AnonGW:~# cat /etc/privoxy/config4
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy4
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8321
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
```

```
buffer-limit 4096
forward-socks5    /               127.0.0.1:9350 .
forward-socks4a   /               127.0.0.1:9350 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

5th

```
AnonGW:~# cat /etc/privoxy/config5
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy5
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8421
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9450 .
forward-socks4a   /               127.0.0.1:9450 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

6th

```
AnonGW:~# cat /etc/privoxy/config6
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy6
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8522
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9550 .
forward-socks4a   /               127.0.0.1:9550 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
```

```
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

7th

```
AnonGW:~# cat /etc/privoxy/config7
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy7
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8623
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9650 .
forward-socks4a   /               127.0.0.1:9650 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

8th

```
AnonGW:~# cat /etc/privoxy/config8
```

```
confdir /etc/privoxy
logdir /var/log/privoxy/privoxy8
actionsfile match-all.action
actionsfile default.action   # Main actions file
actionsfile user-agent.action
actionsfile user.action      # User customizations
filterfile default.filter
logfile logfile
listen-address  127.0.0.1:8724
toggle  1
enable-remote-toggle  0
enable-remote-http-toggle  0
enable-edit-actions 0
enforce-blocks 0
buffer-limit 4096
forward-socks5    /               127.0.0.1:9750 .
forward-socks4a   /               127.0.0.1:9750 .
forward .i2p.xyz 127.0.0.1:4444
forwarded-connect-retries  0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
```

Note:

Before running the demons we have to create log directories:

```
root@AnonGW:~# mkdir /var/log/privoxy/privoxy{2,3,4,5,6,7,8}
```

Those are simple (horrible) scripts to manage multiple demons: **[FIX]**

```
root@AnonGW:~# cat /usr/local/bin/RestartPrivoxy
```

```
#!/bin/bash
/usr/sbin/service privoxy stop ; /bin/sleep 6 ; /usr/sbin/service privoxy start
```

```
root@AnonGW:~# cat /etc/init.d/privoxy
```

```
#! /bin/sh
 ### BEGIN INIT INFO
 # Provides:          privoxy
 # Required-Start:    $local_fs $remote_fs $named $network $time
 # Required-Stop:     $local_fs $remote_fs $named $network $time
 # Default-Start:     2 3 4 5
 # Default-Stop:      0 1 6
 # Short-Description: Privacy enhancing HTTP Proxy
 # Description:       Privoxy is a web proxy with advanced filtering
 #                    capabilities for protecting privacy, filtering
 #                    web page content, managing cookies, controlling
 #                    access, and removing ads, banners, pop-ups and
 #                    other obnoxious Internet junk.
### END INIT INFO
# Do NOT "set -e"
# PATH should only include /usr/* if it runs after the mountnfs.sh script

PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="filtering proxy server"
NAME=privoxy
DAEMON=/usr/sbin/$NAME
PIDFILE=/var/run/$NAME.pid
OWNER=privoxy
CONFIGFILE=/etc/privoxy/config
DAEMON_ARGS="--pidfile $PIDFILE --user $OWNER $CONFIGFILE"
SCRIPTNAME=/etc/init.d/$NAME
LOGDIR=/var/log/privoxy
DEFAULTSFILE=/etc/default/$NAME

[ -x "$DAEMON" ] || exit 0

[ -r $DEFAULTSFILE ] && . $DEFAULTSFILE

if [ ! -d "$LOGDIR" ]; then
    mkdir -m 750 $LOGDIR
    chown $OWNER:adm $LOGDIR
fi

. /lib/init/vars.sh

. /lib/lsb/init-functions

do_start()
{
```

```
    # Return
    #   0 if daemon has been started
    #   1 if daemon was already running
    #   2 if daemon could not be started
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test >
                || return 1
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON -- \
                $DAEMON_ARGS \
                || return 2
    for N in 2 3 4 5 6 7 8
  do
    start-stop-daemon --start --quiet --pidfile /var/run/$NAME$N.pid --exec $DAEMON
  done
    /usr/local/bin/uagen.pl --action-file /etc/privoxy/user-agent.action --action-in


    # Add code here, if necessary, that waits for the process to be ready
    # to handle requests from services started subsequently which depend
    # on this one.  As a last resort, sleep for some time.
}

do_stop()
{
    # Return
    #   0 if daemon has been stopped
    #   1 if daemon was already stopped
    #   2 if daemon could not be stopped
    #   other if a failure occurred
    start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE --nam
    RETVAL="$?"
    [ "$RETVAL" = 2 ] && return 2
    #/usr/bin/killall privoxy 2> /dev/null ; /bin/ps -da | grep uagen.pl  | awk -F '
    /bin/ps -da | grep uagen.pl  | awk -F ' ' '{print $1}' | xargs -L 100 /bin/kill
    /usr/bin/pkill privoxy

        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE2
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE3
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE4
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE5
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE6
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE7
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE8
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE9
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2
    # Wait for children to finish too if this is a daemon that forks
    # and if the daemon is only ever run from this initscript.
```

```
    # If the above conditions are not satisfied then add some other code
    # that waits for the process to drop all resources that could be
    # needed by services started subsequently.  A last resort is to
    # sleep for some time.
    start-stop-daemon --stop --quiet --oknodo --retry=0/30/KILL/5 --exec $DAEMON
    [ "$?" = 2 ] && return 2
    # Many daemons don't delete their pidfiles when they exit.
    rm -f $PIDFILE
    rm -f $PIDFILE[2-8]
    return "$RETVAL"
}

do_reload() {
    #
    # If the daemon can reload its configuration without
    # restarting (for example, when it is sent a SIGHUP),
    # then implement that here.
    #
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE2 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE3 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE4 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE5 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE6 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE7 --name $NAME
    start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE8 --name $NAME
    return 0
}


case "$1" in
  start)
    if [ "$RUN_DAEMON" = "no" ]; then
            [ "$VERBOSE" != no ] && log_warning_msg "Not starting $DESC (disabled in
            exit 0
    fi

    [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
    do_start
    case "$?" in
            0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
            2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;

  restart|force-reload)
    #
    # If the "reload" option is implemented then remove the
    # 'force-reload' alias
    #
    if [ "$RUN_DAEMON" = "no" ]; then
            [ "$VERBOSE" != no ] && log_warning_msg "Not restarting $DESC (disabled
            exit 0
    fi

    log_daemon_msg "Restarting $DESC" "$NAME"
    do_stop
    case "$?" in
      0|1)
            do_start
            case "$?" in
```

```
                    0) log_end_msg 0 ;;
                    1) log_end_msg 1 ;; # Old process is still running
                    *) log_end_msg 1 ;; # Failed to start
            esac
            ;;
      *)
            # Failed to stop
            log_end_msg 1
            ;;
    esac
    ;;
  status)
       status_of_proc "$DAEMON" "$NAME"
       exit $?
       ;;

  *)
    #echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
    echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload|status}" >&2
    exit 3
    ;;
esac
```

Posted gio 16 mag 2013 13:29:45 UTC

Tags: privoxy

squid3

The original idea to use multiple balanced instances of TOR was borrowed from this article.

Below is the configuration file **squid.conf**:

```
AnonGW:~# cat /etc/squid3/squid.conf
```

```
visible_hostname DEEP_WEB_PROXY
cache_mgr anonymous
client_db on
detect_broken_pconn on
dns_defnames on
dns_retransmit_interval 2 seconds
dns_timeout 5 minutes
forwarded_for off
half_closed_clients off
httpd_suppress_version_string on
ignore_unknown_nameservers on
pipeline_prefetch on
retry_on_error on
strip_query_terms off
uri_whitespace strip

icp_port 0
icp_access deny all
redirect_rewrites_host_header off
dns_nameservers 127.0.0.1
http_port $DefaultGW:3128 intercept
```

```
forward_timeout 90 seconds
connect_timeout 90 seconds
read_timeout 220 seconds
request_timeout 90 seconds
peer_connect_timeout 90 seconds
persistent_request_timeout 4 minutes
client_lifetime 20 hours

ident_lookup_access deny all

acl apache rep_header Server ^Apache

cache_mem 512 MB
cache_log /var/spool/cache/squid3/log/cache.log
cache_access_log /var/spool/cache/squid3/log/access.log
cache_store_log /var/spool/cache/squid3/log/store.log
cache_swap_log /var/spool/cache/squid3/log/swap.log
logfile_rotate 10

# acl QUERY urlpath_regex cgi-bin \?
# acl ban url_regex "/etc/squid3/bad-world.squid"
acl localnet src $IntNet $VPN
acl localhost src 127.0.0.1/32
acl loc-serv dst 127.0.0.1 $VPN
acl manager proto cache_object
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 2083 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 2083 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
acl FTP proto FTP
never_direct allow all
always_direct allow FTP
acl bad_url dstdomain "/etc/squid3/bad-sites.squid"
acl malware_domains url_regex "/etc/squid3/Malware-domains.txt"
acl ads dstdom_regex "/etc/squid3/ad_block.txt"
http_access deny ads
no_cache deny loc-serv
# no_cache deny QUERY
cache_dir ufs /var/spool/cache/squid3 8192 16 256
maximum_object_size 10 MB
store_avg_object_size 500 KB
log_fqdn on
# max connections per ip
acl maxuserconn src 127.0.0.0/8 $IntNet $VPN
acl limitusercon maxconn 500
http_access deny maxuserconn limitusercon

refresh_pattern ^(ht|f)tp://.*ubuntu.*/Packages\.(bz2|gz|diff/Index)$   0       0%
refresh_pattern ^(ht|f)tp://.*ubuntu.*/Release(\.gpg)?$                 0       0%
refresh_pattern ^(ht|f)tp://.*ubuntu.*/Sources\.(bz2|gz|diff/Index)$    0       0%
refresh_pattern ^(ht|f)tp://.*ubuntu.*/Translation-it\.bz2)$         0       0%
```

```
hierarchy_stoplist cgi-bin ?
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

cache_peer localhost parent 8118 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost2 parent 8129 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost3 parent 8230 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost4 parent 8321 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost5 parent 8421 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost6 parent 8522 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost7 parent 8623 0 round-robin no-query no-digest no-netdb-exchange
cache_peer localhost8 parent 8724 0 round-robin no-query no-digest no-netdb-exchange

always_direct allow loc-serv
cachemgr_passwd disable all
uri_whitespace encode
http_access allow manager localhost localnet
http_access deny bad_url
http_access deny malware_domains
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny manager
acl apache rep_header Server ^Apache
log_icp_queries off
buffered_logs on
coredump_dir /tmp

## redirector
acl my_url dstdomain https://duckduckgo.com
redirector_access allow my_url
redirect_children 1
redirect_program /etc/squid3/squid_redirector.pl

## methods allowed
acl Safe_method method CONNECT GET HEAD POST
http_access deny !Safe_method
always_direct allow localhost
never_direct allow localnet
shutdown_lifetime 0 seconds

reply_header_access From deny all
reply_header_access Referer deny all
reply_header_replace Referer unknown
reply_header_access Server deny all
reply_header_access User-Agent deny all
reply_header_replace User-Agent SecretBrowser/5.0 (Commodore64; en)
reply_header_access X-Forwarded-For deny all
reply_header_replace X-Forwarded-For 11.11.11.11
reply_header_access WWW-Authenticate deny all
reply_header_access Link deny all
reply_header_access Via deny all
reply_header_replace Via 11.11.11.11
reply_header_access Allow allow all
reply_header_access Authorization allow all
reply_header_access WWW-Authenticate allow all
reply_header_access Proxy-Authorization allow all
reply_header_access Proxy-Authenticate allow all
reply_header_access Cache-Control allow all
reply_header_access Content-Encoding allow all
```

```
reply_header_access Content-Length allow all
reply_header_access Content-Type allow all
reply_header_access Date allow all
reply_header_access Expires allow all
reply_header_access Host allow all
reply_header_access If-Modified-Since allow all
reply_header_access Last-Modified allow all
reply_header_access Location allow all
reply_header_access Pragma allow all
reply_header_access Accept allow all
reply_header_access Accept-Charset allow all
reply_header_access Accept-Encoding allow all
reply_header_access Accept-Language allow all
reply_header_access Content-Language allow all
reply_header_access Mime-Version allow all
reply_header_access Retry-After allow all
reply_header_access Title allow all
reply_header_access Connection allow all
reply_header_access Cookie allow all
acl cookie_allow dstdomain "/etc/squid3/cookie_allow"
reply_header_access Set-Cookie allow cookie_allow
reply_header_access Set-Cookie deny all
#reply_header_access Set-Cookie allow all
reply_header_access All deny all
via off
forwarded_for delete
```

Note:

We have to create a new cache directory in the encrypted device:

```
mkdir /var/spool/cache/squid3/ && chown proxy:proxy /var/spool/cache/squid3
```

This configuration may be redundant in some directives when used in chain with Privoxy.

We have to pay attention to the **cache_peer** directives: we need *N* directives for all *N* Privoxy parent proxies (here *N* = 8) and in **/etc/hosts**:

```
127.0.0.1 localhost
127.0.0.1 localhost2
127.0.0.1 localhost3
127.0.0.1 localhost4
127.0.0.1 localhost5
127.0.0.1 localhost6
127.0.0.1 localhost7
127.0.0.1 localhost8
```

where *localhostN* are the names referred to the number of the N clustered parent proxies.

To log in 'tormail': **/etc/squid3/cookie_allow**

```
AnonGW:~# cat /etc/squid3/cookie_allow
.jhiwjjlqpyawmpjx.onion
```

Cron-script to get the list in **/etc/squid3/Malware-domains.txt** and **/etc/squid3/ad_block.txt**:

```
AnonGW:~# cat /usr/local/bin/update-domains.sh
```

```
#!/bin/bash
mkdir /tmp/miep
touch /tmp/malware.tmp
for i in `/usr/bin/python /usr/local/bin/ml.py`; do mkdir -p /tmp/miep/$i; done
ls /tmp/miep/ > /tmp/malware.tmp
diff -a /etc/squid3/Malware-domains.txt /tmp/malware.tmp >> /tmp/diff.tmp
patch /etc/squid3/Malware-domains.txt /tmp/diff.tmp
rm /tmp/malware.tmp /tmp/diff.tmp
rm -rf /tmp/miep
wget -O /etc/squid3/ad_block.txt 'http://pgl.yoyo.org/adservers/serverlist.php?hostf
echo "keke done!"

AnonGW:~# cat /usr/local/bin/ml.py
    import re
import urllib2

# download website
request = urllib2.Request('http://www.malwaredomainlist.com/mdl.php?sort=Domain&asco
opener = urllib2.build_opener()
try:
    content = opener.open(request).read()
except:
    print "FAIL!"
    content = ""

re_tr = re.compile('<tr(.*?)</tr>')
re_td = re.compile('<td>(.*?)</td>')

for tr in re_tr.findall(content):
    try:
        tds = re_td.findall(tr)
        domain = tds[1]
        ip = tds[2]

        to_out = domain
        if domain == '-':
            to_out = ip

        print to_out.replace("<wbr>", "")
    except:
        pass
```

Posted mar 14 mag 2013 00:09:37 UTC

Tags: squid3

testing environment

The testing environment has this *variables* :

$[FIX] ==> something of horrible to fix urgently - any comment with a suggestion will be appreciated.

N = *n* = number of concurrent Privoxy and TOR processes = "**8**" in this proof of concept.

Networking

```
External NIC = $ExtNIC ==> Bridge on $ExtBr

Internal NIC = $IntNIC ==> Bridge on $IntBr

IP Address External NIC = $ExtBr_IP = $Ext_IP

IP Address Internal NIC = $IntBr_IP = $DefaultGW (Default Gateway)

OpenVPN network = $VPN ==> NIC = $tun

LAN = $IntNet | KVM network = $vir_net | LXC network = $lxc_net

KVM bridged NIC = $virbr | LXC bridged NIC = $lxcbr
```

Note: the network bridge is useful when you isolate processes or systems with LXC or KVM (ex.: eepsites or hidden services).

Storage (simple layout)

```
root@AnonGW:~# mdadm --detail /dev/md{0,1}

/dev/md0:
        Version : 1.2
  Creation Time : Fri May  3 16:01:55 2013
     Raid Level : raid1
     Array Size : 498676 (487.07 MiB 510.64 MB)
  Used Dev Size : 498676 (487.07 MiB 510.64 MB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Tue May 14 18:14:53 2013
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           Name : AnonGW:0  (local to host AnonGW)
           UUID : 7341b880:a0dd7a1c:3336d224:432f6390
         Events : 20

    Number   Major   Minor   RaidDevice State
       0       8        1        0      active sync   /dev/sda1
       1       8       17        1      active sync   /dev/sdb1
/dev/md1:
        Version : 1.2
  Creation Time : Fri May  3 16:02:07 2013
     Raid Level : raid1
     Array Size : 77648824 (74.05 GiB 79.51 GB)
  Used Dev Size : 77648824 (74.05 GiB 79.51 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Tue May 14 18:14:59 2013
          State : clean
 Active Devices : 2
```

```
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           Name : AnonGW:1  (local to host AnonGW)
           UUID : 63fa5545:4f510647:0f78b18f:e02cb327
         Events : 30

    Number   Major   Minor   RaidDevice State
       0       8        5        0      active sync   /dev/sda5
       1       8       21        1      active sync   /dev/sdb5

root@AnonGW:~# pvs

PV         VG     Fmt  Attr PSize  PFree
/dev/md1   AnonGW lvm2 a-   74,05g 12,84g

root@AnonGW:~# vgs

VG     #PV #LV #SN Attr   VSize  VFree
AnonGW   1   7   0 wz--n- 74,05g 12,84g

root@AnonGW:~# lvs

LV    VG     Attr   LSize   Origin Snap%  Move Log Copy%  Convert
cache AnonGW -wi-ao  10,00g
home  AnonGW -wi-ao  18,62g
root  AnonGW -wi-ao   3,72g
swap  AnonGW -wi-ao   3,72g
tmp   AnonGW -wi-ao 952,00m
usr   AnonGW -wi-ao   5,59g
var   AnonGW -wi-ao  18,62g
```

/var/spool/cache/ is mounted on an encrypted device by a random key: every reboot all data on this volume will be destroyed permanently

```
root@AnonGW:~# lsblk

sda                                 8:0    0  74,5G  0 disk
├─sda1                              8:1    0   487M  0 part
│ └─md0                             9:0    0   487M  0 raid1 /boot
├─sda2                              8:2    0     1K  0 part
└─sda5                              8:5    0  74,1G  0 part
  └─md1                             9:1    0  74,1G  0 raid1
    ├─AnonGW-cache (dm-8)         252:8    0    10G  0 lvm
    │ └─AnonGW-cache_crypt (dm-9) 252:9    0    10G  0 crypt /var/spool/cache
    ├─AnonGW-root (dm-0)          252:0    0   3,7G  0 lvm   /
    ├─AnonGW-tmp (dm-1)           252:1    0   952M  0 lvm
    │ └─AnonGW-tmp_crypt (dm-7)   252:7    0   952M  0 crypt /tmp
    ├─AnonGW-usr (dm-2)           252:2    0   5,6G  0 lvm   /usr
    ├─AnonGW-swap (dm-3)          252:3    0   3,7G  0 lvm
    │ └─AnonGW-swap_crypt (dm-6)  252:6    0   3,7G  0 crypt [SWAP]
    ├─AnonGW-home (dm-4)          252:4    0  18,6G  0 lvm   /home
    └─AnonGW-var (dm-5)           252:5    0  18,6G  0 lvm   /var
sdb                                 8:16   0  74,5G  0 disk
├─sdb1                              8:17   0   487M  0 part
│ └─md0                             9:0    0   487M  0 raid1 /boot
├─sdb2                              8:18   0     1K  0 part
└─sdb5                              8:21   0  74,1G  0 part
  └─md1                             9:1    0  74,1G  0 raid1
```

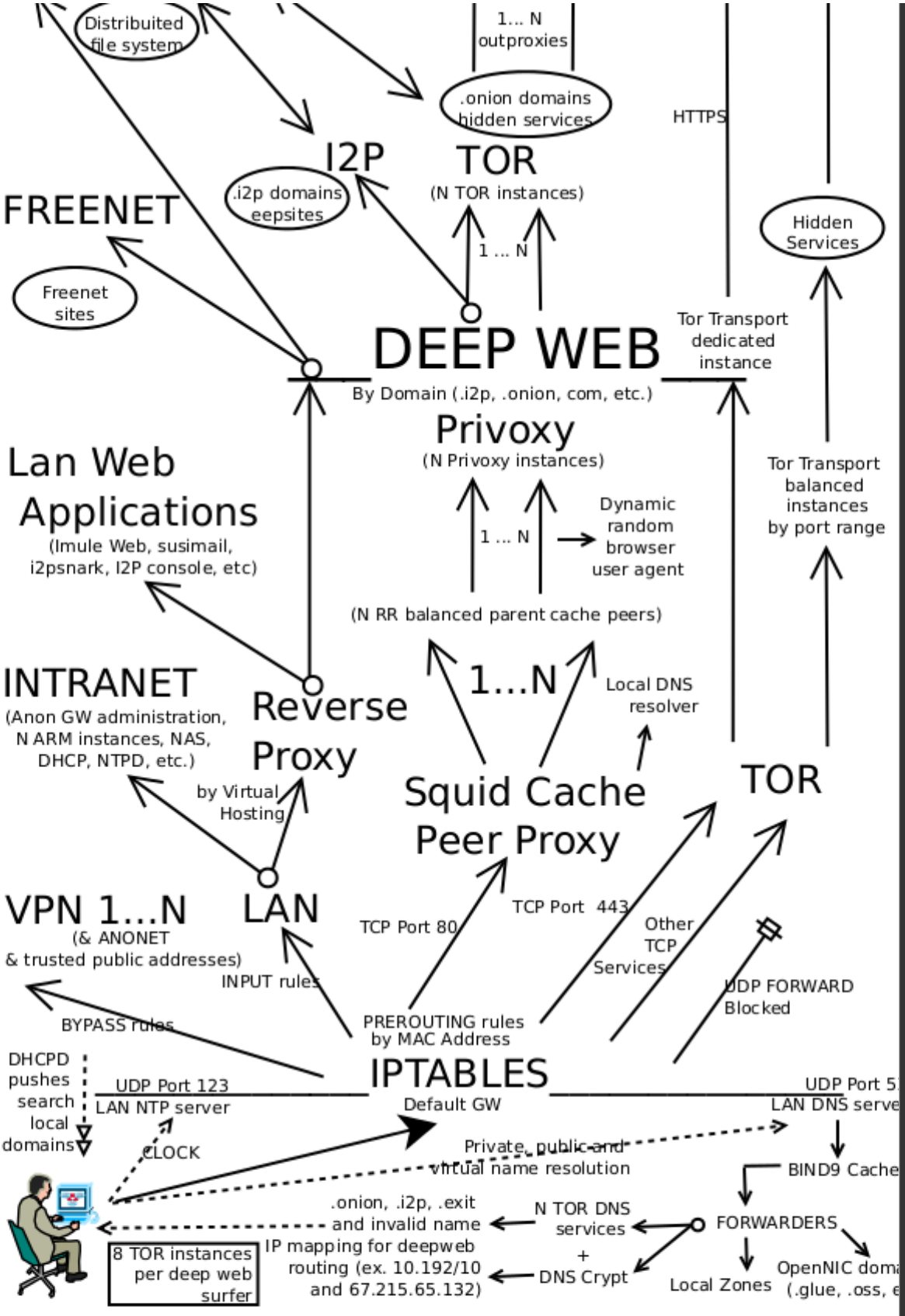
```
    ├─AnonGW-cache (dm-8)          252:8    0    10G  0 lvm
    │ └─AnonGW-cache_crypt (dm-9) 252:9    0    10G  0 crypt /var/spool/cache
    ├─AnonGW-root (dm-0)           252:0    0   3,7G  0 lvm   /
    ├─AnonGW-tmp (dm-1)            252:1    0   952M  0 lvm
    │ └─AnonGW-tmp_crypt (dm-7)    252:7    0   952M  0 crypt /tmp
    ├─AnonGW-usr (dm-2)            252:2    0   5,6G  0 lvm   /usr
    ├─AnonGW-swap (dm-3)           252:3    0   3,7G  0 lvm
    │ └─AnonGW-swap_crypt (dm-6)   252:6    0   3,7G  0 crypt [SWAP]
    ├─AnonGW-home (dm-4)           252:4    0  18,6G  0 lvm   /home
    └─AnonGW-var (dm-5)            252:5    0  18,6G  0 lvm   /var

root@AnonGW:~# findmnt

TARGET                       SOURCE                                      FSTYPE
/                            /dev/mapper/AnonGW-root                     ext4
├─/sys                       sysfs                                       sysfs
│ ├─/sys/fs/fuse/connections                                            fusectl
│ ├─/sys/kernel/debug                                                   debugfs
│ └─/sys/kernel/security                                                securityfs
├─/proc                      proc                                        proc
│ └─/proc/sys/fs/binfmt_misc binfmt_misc                                 binfmt_misc
├─/dev                       udev                                        devtmpfs
│ └─/dev/pts                 devpts                                      devpts
├─/run                       tmpfs                                       tmpfs
│ ├─/run/lock                                                           tmpfs
│ └─/run/shm                                                            tmpfs
├─/usr                       /dev/mapper/AnonGW-usr                      ext4
├─/home                      /dev/mapper/AnonGW-home                     ext4
│ └─/home/anonymous          /home/anonymous/.Private                    ecryptfs
├─/var                       /dev/mapper/AnonGW-var                      ext4
│ ├─/var/spool/cache         /dev/mapper/AnonGW-cache_crypt              ext2
│ ├─/var/mail                /dev/mapper/AnonGW-cache_crypt[/mail]       ext2
│ └─/var/log/squid3          /dev/mapper/AnonGW-cache_crypt[/squid3/log] ext2
├─/boot                      /dev/md0                                    ext3
└─/tmp                       /dev/mapper/AnonGW-tmp_crypt                ext2
```

Posted lun 13 mag 2013 23:59:43 UTC

Tags: POC

deep web proxy

This is the general scheme:


Distributed file system
1... N outproxies
.onion domains hidden services
HTTPS
I2P
TOR
(N TOR instances)
.i2p domains eepsites
FREENET
1 ... N
Hidden Services
Freenet sites
DEEP WEB
By Domain (.i2p, .onion, com, etc.)
Tor Transport dedicated instance
Privoxy
(N Privoxy instances)
Lan Web Applications
(Imule Web, susimail, i2psnark, I2P console, etc)
1 ... N
Dynamic random browser user agent
Tor Transport balanced instances by port range
(N RR balanced parent cache peers)
INTRANET
(Anon GW administration, N ARM instances, NAS, DHCP, NTPD, etc.)
Reverse Proxy
by Virtual Hosting
1...N
Local DNS resolver
Squid Cache Peer Proxy
TOR
VPN 1...N
(& ANONET & trusted public addresses)
LAN
INPUT rules
TCP Port 80
TCP Port 443
Other TCP Services
UDP FORWARD Blocked
BYPASS rules
PREROUTING rules by MAC Address
DHCPD pushes search local domains
IPTABLES
Default GW
UDP Port 123
LAN NTP server
UDP Port 5
LAN DNS serve
CLOCK
Private, public and virtual name resolution
BIND9 Cache
.onion, .i2p, .exit and invalid name
IP mapping for deepweb routing (ex. 10.192/10 and 67.215.65.132)
N TOR DNS services
+
DNS Crypt
FORWARDERS
Local Zones
OpenNIC doma (.glue, .oss, e
8 TOR instances per deep web surfer

Here the data flows of our anonymous surfer Jon Do...

Requirements:

1. A box with a dual core processor, two network interfaces and two hard drives.
2. Ubuntu server 12.04 64 bit.
3. RAID 1 (if paranoia/comfort > 1 then whole disk encrypted) and/or: random key encrypted swap, random key encrypted tmp logical volume and random key encrypted cache volume.
4. ISC DHCP, Bind9, DnsCrypt, NTPD, Squid3, N Privoxy instances, N TOR instances (where N is 8 * deep web surfer), I2P, Tahoe-LAFS-I2P, Freenet, Apache2, OpenVPN
5. Patience and passion

How it works:

1. Jon Do adds in the AnonGW the MAC address of his new laptop (or tablet or smartphone or anything else with the capability of web browsing and a known physical address...)
2. The laptop broadcasts its presence in the LAN, asking networking settings ==> the DHCP server (isc-dhcp) instructs which IP address, default gateway, NTP server, DNS server and which DNS domain to search are needed. The domain will be the local LAN domain and the VPN's domains (office, datacentre, etc.).
3. he laptop starts to synchronize its clock with the clock of the AnonGW. Jon Do searches "http://www.debian.org" in his browser and its DNS client queries AnonGW DNS server to resolve it. If the answer is not cached, the DNS server (bind9) forwards the query via TOR network or, as last attempt, via DnsCrypt on OpenDNS
4. HTTP request is intercepted by the cache proxy (Squid3) and "round robin" balanced over the N (ex. 8) Privoxy parent peers. The browser user agent are dinamically modified by a script (uagent.py) and forwarded via N (ex: 8) TOR instances.
5. Jon Do searches https://www.eff.org and open a ssh session on a public server: both connections are "transported" by two different instances of TOR.
6. Jon Do opens a new tab in his web browser and searches his Tor Mail "http://jhiwjjlqpyawmpjx.onion": the DNS server forwards directly to the TOR DNS service that resolves the ".onion" domain with a private address (ex: 10.192.0.1). HTTP request is intercepted by Squid and forwarded to Privoxy: it recognizes the "dot onion" domain and forwards it in the TOR network to find the "hidden service".
7. Jon Do opens again a new tab and searches "http://killyourtv.i2p.xyz/debian/": the DNS server forwards directly to the TOR DNS service that resolves the ".i2p.xyz" domain with a private address (ex: 10.192.0.2). HTTP request is always intercepted by Squid and forwarded to Privoxy: it recognizes the "dot i2p" domain and forwards it in the I2P network to find the "eepsite".
8. Entering only "i2p" in his address bar he has access to the i2p console or to a invisible torrent client on "https://i2p/i2psnark/": the DNS find in its local zones and resolves it with the AnonGW IP and the HTTP/HTTPS request is authenticated by the Apache2 I2P.JonDolocaldomain virtual host and forwarded via its reverse proxy to the i2p router.

9. Entering only "tahoe" in his address bar he has access to the Tahoe-LAFS welcome page: Jon Do can find the KYTV's Tahoe-LAFS debian repository on:

```
https://tahoe/uri/URI:DIR2-RO:mvp3so6kemo6fn6abddzjthnuu:jjg475us6hbmya3ccy
```

1. Entering only "freenet" in his address bar he has access to the Freenet console and can read the Toad's Flog on:

```
https://freenet/freenet:USK@yGvITGZzrY1vUZK-4AaYLgcjZ7ysRqNTMfdcO8gS-LY,-al
```

1. Jon Do can access to the LAN of his office via VPN: he has configured the DNS zone and the VPN service on his Anonymous Gateway. Anonet is the same concept: he has configured a VPN but the DNS forwards the ".ano" queries to the Anonet DNS.

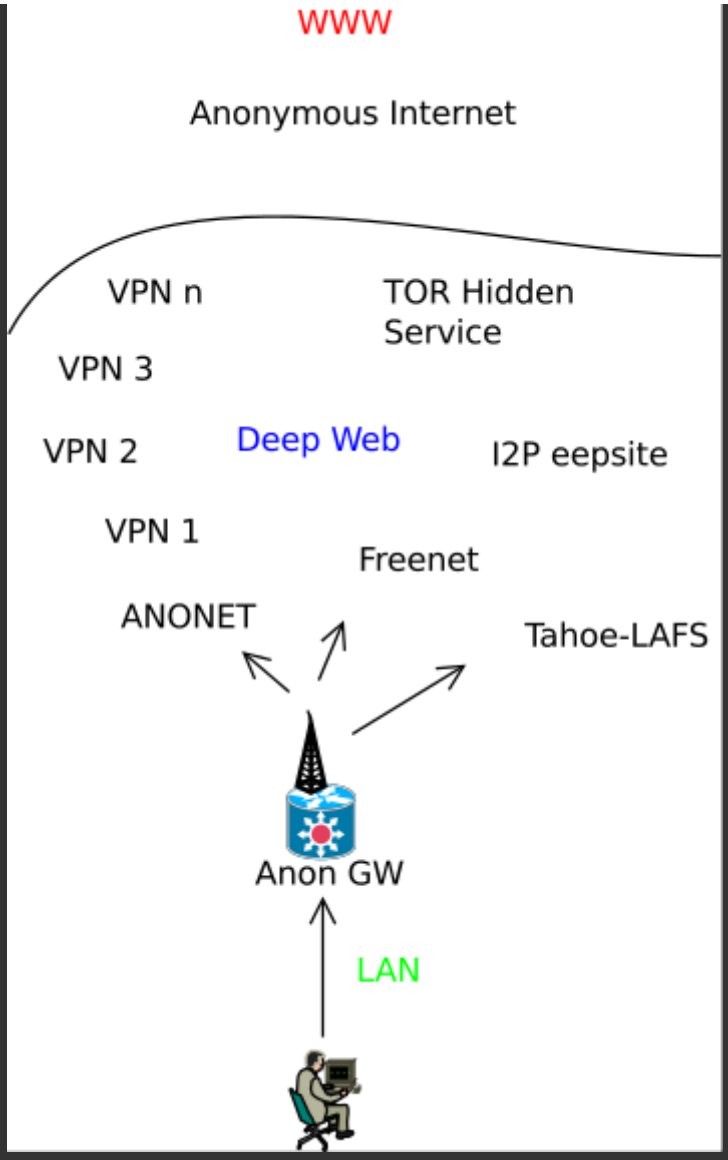
2. Jon Do needs to do a different research and removes his MAC address from the "Deep Web Proxy rules" from the AnonGW and search "http://grep.geek": the DNS server forwards the queries to the OpenNIC DNS servers. He can surf free TLDs and normal clearnet without anonymization.

Posted mer 24 apr 2013 13:26:45 UTC

Tags: POC

proof of concept

This is the idea...



One or more persons can gain internet resources in the most secure, fast and handy method via a server gateway that, via a proxy system, can route transparently all the traffic from layer 2 to layer 7 to reach local and remote services, in the clearnet or in the opennet/darknet.

Of course the best method for web anonymization is the use, for example, of the TOR bundle or Tails; but I think this is an excellent compromise for daily web surfing in west democratic (ROTFL) countries compared to the top paranoia in the "they can kill me" dictatorships.

Of course a good method to use the AnonGW is surfing with the TOR browser in transparent mode. The users don't need to install anything on their workstations and they can free resources and decrease complexity in their boxes (no java running on their laptops 🙂)

The AnonGW provides:

- DHCP service ==> automatically dictates the network rules to use the deep web proxy.
- Time service (NTP server) ==> syncronizes LAN clocks, no UDP leaks out of the LAN.
- DNS service (bind9) ==> (I) manages local zones for LAN services, VPN networks and reverse proxy (II) forwards the queries to the TOR network and, for a full answer, to OpenDNS via DNSCrypt.

- filtering service (iptables) ==> activate/deactivate the deep web proxy filtering mac address (layer 2 oriented).
- firewalling all protocols except toward VPN's and Anonet (layer 3 oriented).
- routing tcp via proxies (layer 4 oriented).
- web traffic managed through HTTP headers via direct cache, reverse and filtering web proxies (layer 7 oriented).
- access anonymously to web (clearnet and hidden services) through TOR: (I) fast HTTP via a round robin clustered cache proxy (Squid3) over multiple privoxy and TOR instances (II) HTTPS and other TCP services directly via TOR (transport mode).
- access to I2P resources: (I) eeepsites via Squid and Privoxy (II) I2P web clients via reverse proxy (Apache2 virtual host).
- access to Freenet resources: via reverse proxy (virtual host).
- access to Tahoe-LAFS storage grids in I2P (or TOR or anything else): via reverse proxy (virtual host) or via SSHFS.
- access to OpenNIC domains but note: only few exit nodes resolve these domains.

Posted mar 23 apr 2013 16:59:33 UTC

Tags: POC

abstract

The objective of this blog is to be a tutorial to realize a high speed gateway to anonymize the Internet traffic and routing it to the main free spaces in the Internet.

The idea has born from the need to force myself to improve my privacy in my daily web surfing without renouncing too much to the performance and to have always available, with any workstation, the most important anonymous networks and services.

The Anonymous Gateway is a set of open source software customized to work together on a standard *nix platform.

Every post will be a "how to" configure every client and every server to reach the goal: a centralized set of tools for anonymization.

Every services is a world of knowledge and I will be glad if anyone will want to contribute to improve performance, security or anything else.

It would be nice developing a "simple web admin interface"; now I'm using a simple aggregation of links to the specific consoles or to the virtual hosts: I'm a sysadmin not a developer.

I hope this blog will also be useful to everyone looking for tutorials with working configurations of the most used open source software in server systems.

I used an Ubuntu Gnu/Linux server 12.04 (LTS) with apparmor enabled and with all its standard repository packages, included I2P repository for "Precise" mantained by KillYourTV (thank you!).

Posted gio 18 apr 2013 08:10:04 UTC

Tags: abstract intro

This blog is powered by ikiwiki. -- Contacts: zoidberg@i2pmail.org

---

Last edited ven 26 lug 2013 21:10:18 UTC