From: Andrew Lewman, Executive Director
To: The Tor Community
Date: September 15, 2011

This report documents progress in August 2011.

New releases, new hires, new funding

New Releases

1. On August 10th, we released a new version of Vidalia, the graphical controller interface for Tor, version 0.2.13.

```
0.2.13  10-Aug-2011
  o Add a way to bootstrap Tor's torrc file (copy the torrc to a given
    directory before Vidalia starts) so that packages such as
    Bridge-by-default portable bundles for OSX don't violate the
    directory structure of the operating system. Fixes bug 2821.
  o Add the proper CA Certificates so that the "Find Bridges" button
    works again. Fixes bug 2835.
  o Update the useful links help page. Fixes bug 2809.
  o Reintegrate Breakpad, and make it available in platforms other than
    Windows. Resolves bug 2105.
  o Fix bandwidth assigned to relays on the Network Map. A lot of relays
    are displaying an erroneous bandwidth and since they are ordered by that
    value in the Network Map, it leads to confusion. Vidalia now
    specifies the bandwidth as the minimum of the three possible values (burst,
    average and observed). Fixes bug 2744.
  o Fix layouts in the configuration panel to make them look seamlessly
    across all platforms.
  o Add -no-remote parameter to Firefox so it allows another instance of
    non-TBB Firefox. Fixes bug 2254.
  o Add the possibility of changing the torrc path while Tor hasn't
    started. Fixes bug 3109.
  o Make the fact that bridges don't need a DirPort setting more clear
    by removing the content of the field when disabling it. Fixes bug 3119.
  o Improve command line parameter handling. Resolves bug 2965.
  o Fix layout in BandwidthGraph to display labels correctly in every
    language. Fixes bug 2500.
  o Updates README.debs to reflect the change in the packaging now that
```

```
    Vidalia uses Git. Fixes bug 3668.
  o Add a way to use the autoconfiguration for ControlPort and
    SocksPort.  Tor can now autoconfigure Control and Socks Ports when
    the default ones are in use. This makes it easier to run several
    different instances of TBB at the same time. Resolves bug 3077.
  o Provide the necessary fields (Control password, ControlPort) to let
    TorButton NEWNYM. Vidalia provides these in env vars when it
    launches the Firefox instance. Resolves bug 2659.
```

2. On August 17th, we released the second and final release candidate for the 0.2.2.x series, Tor 0.2.2.31-rc.

```
Changes in version 0.2.2.31-rc - 2011-08-17
  o Major bugfixes:
    - Remove an extra pair of quotation marks around the error
      message in control-port STATUS_GENERAL BUG events. Bugfix on
      0.1.2.6-alpha; fixes bug 3732.
    - If we’re configured to write our ControlPorts to disk, only write
      them after switching UID and creating the data directory. This way,
      we don’t fail when starting up with a nonexistent DataDirectory
      and a ControlPortWriteToFile setting based on that directory.
      Fixes bug 3747; bugfix on Tor 0.2.2.26-beta.

  o Minor features:
    - Update to the August 2 2011 Maxmind GeoLite Country database.

  o Minor bugfixes:
    - Allow GETINFO fingerprint to return a fingerprint even when
      we have not yet built a router descriptor. Fixes bug 3577;
      bugfix on 0.2.0.1-alpha.
    - Write several files in text mode, on OSes that distinguish text
      mode from binary mode (namely, Windows). These files are:
      ’buffer-stats’, ’dirreq-stats’, and ’entry-stats’ on relays
      that collect those statistics; ’client_keys’ and ’hostname’ for
      hidden services that use authentication; and (in the tor-gencert
      utility) newly generated identity and signing keys. Previously,
      we wouldn’t specify text mode or binary mode, leading to an
      assertion failure. Fixes bug 3607. Bugfix on 0.2.1.1-alpha (when
      the DirRecordUsageByCountry option which would have triggered
      the assertion failure was added), although this assertion failure
      would have occurred in tor-gencert on Windows in 0.2.0.1-alpha.
    - Selectively disable deprecation warnings on OS X because Lion
      started deprecating the shipped copy of openssl. Fixes bug 3643.
    - When unable to format an address as a string, report its value
      as "???" rather than reusing the last formatted address. Bugfix
      on 0.2.1.5-alpha.
```

3. On August 19, updated Tor Browser Bundles.

   ```
   Windows bundle
   1.3.27: Released 2011-08-19

       Update Firefox to 3.6.20
       Update Libevent to 2.0.13-stable
       Update HTTPS-Everywhere to 1.0.0development.5

   OS X bundle
   1.0.23: Released 2011-08-19

       Update Tor to 0.2.2.31-rc
       Update Firefox to 3.6.20
       Update Libevent to 2.0.13-stable
       Update NoScript to 2.1.2.6
       Update HTTPS-Everywhere to 1.0.0development.5
       Remove BetterPrivacy until we can figure out how to make it safe in
       all bundles (see #3597)

   Linux bundle
   1.1.13: Released 2011-08-19

       Update Tor to 0.2.2.31-rc
       Update Firefox to 3.6.20
       Update Libevent to 2.0.13-stable
       Update NoScript to 2.1.2.6
       Update HTTPS-Everywhere to 1.0.0development.5
       Remove BetterPrivacy until we can figure out how to make it safe in
       all bundles (see #3597)
   ```

4. On August 21, updated Tor Browser Bundles.

   ```
   Tor Browser Bundle (2.2.31-1) alpha; suite=all
       Update Tor to 0.2.2.31-rc
       Update Firefox to 6.0
       Update Libevent to 2.0.13-stable
       Update NoScript to 2.1.2.6
       Update HTTPS Everywhere to 1.0.0development.5
       Remove BetterPrivacy until we can figure out how to make it safe in
       all bundles (see #3597)
   ```

5. On August 26th, we released an updated version of Vidalia, 0.2.14.

   ```
   0.2.14  26-Aug-2011
     o Make the AutoPort setting default to false, so that it doesn't
   ```

```
    break backwards compatibility for people that aren't using Vidalia
    inside Tor Browser Bundle.
```

6. On August 27th, we released a new stable Tor, version 0.2.2.32.

```
The Tor 0.2.2 release series is dedicated to the memory of Andreas
Pfitzmann (1958-2010), a pioneer in anonymity and privacy research,
a founder of the PETS community, a leader in our field, a mentor,
and a friend. He left us with these words: "I had the possibility
to contribute to this world that is not as it should be. I hope I
could help in some areas to make the world a better place, and that
I could also encourage other people to be engaged in improving the
world. Please, stay engaged. This world needs you, your love, your
initiative -- now I cannot be part of that anymore."

Tor 0.2.2.32, the first stable release in the 0.2.2 branch, is finally
ready. More than two years in the making, this release features improved
client performance and hidden service reliability, better compatibility
for Android, correct behavior for bridges that listen on more than
one address, more extensible and flexible directory object handling,
better reporting of network statistics, improved code security, and
many many other features and bugfixes.

https://www.torproject.org/download/download

Changes in version 0.2.2.32 - 2011-08-27
  o Major features (client performance):
    - When choosing which cells to relay first, relays now favor circuits
      that have been quiet recently, to provide lower latency for
      low-volume circuits. By default, relays enable or disable this
      feature based on a setting in the consensus. They can override
      this default by using the new "CircuitPriorityHalflife" config
      option. Design and code by Ian Goldberg, Can Tang, and Chris
      Alexander.
    - Directory authorities now compute consensus weightings that instruct
      clients how to weight relays flagged as Guard, Exit, Guard+Exit,
      and no flag. Clients use these weightings to distribute network
      load more evenly across these different relay types. The weightings
      are in the consensus so we can change them globally in the future.
      Extra thanks to "outofwords" for finding some nasty security bugs
      in the first implementation of this feature.

  o Major features (client performance, circuit build timeout):
    - Tor now tracks how long it takes to build client-side circuits
      over time, and adapts its timeout to local network performance.
      Since a circuit that takes a long time to build will also provide
      bad performance, we get significant latency improvements by
      discarding the slowest 20% of circuits. Specifically, Tor creates
      circuits more aggressively than usual until it has enough data
      points for a good timeout estimate. Implements proposal 151.
    - Circuit build timeout constants can be controlled by consensus
      parameters. We set good defaults for these parameters based on
      experimentation on broadband and simulated high-latency links.
    - Circuit build time learning can be disabled via consensus parameter
      or by the client via a LearnCircuitBuildTimeout config option. We
      also automatically disable circuit build time calculation if either
      AuthoritativeDirectory is set, or if we fail to write our state
      file. Implements ticket 1296.

  o Major features (relays use their capacity better):
    - Set SO_REUSEADDR socket option on all sockets, not just
      listeners. This should help busy exit nodes avoid running out of
      useable ports just because all the ports have been used in the
      near past. Resolves issue 2850.
    - Relays now save observed peak bandwidth throughput rates to their
      state file (along with total usage, which was already saved),
      so that they can determine their correct estimated bandwidth on
      restart. Resolves bug 1863, where Tor relays would reset their
      estimated bandwidth to 0 after restarting.
    - Lower the maximum weighted-fractional-uptime cutoff to 98%. This
      should give us approximately 40-50% more Guard-flagged nodes,
      improving the anonymity the Tor network can provide and also
      decreasing the dropoff in throughput that relays experience when
      they first get the Guard flag.
    - Directory authorities now take changes in router IP address and
      ORPort into account when determining router stability. Previously,
      if a router changed its IP or ORPort, the authorities would not
      treat it as having any downtime for the purposes of stability
      calculation, whereas clients would experience downtime since the
      change would take a while to propagate to them. Resolves issue 1035.
    - New AccelName and AccelDir options add support for dynamic OpenSSL
      hardware crypto acceleration engines.

  o Major features (relays control their load better):
    - Exit relays now try harder to block exit attempts from unknown
      relays, to make it harder for people to use them as one-hop proxies
      a la tortunnel. Controlled by the refuseunknownexits consensus
      parameter (currently enabled), or you can override it on your
      relay with the RefuseUnknownExits torrc option. Resolves bug 1751;
      based on a variant of proposal 163.
    - Add separate per-conn write limiting to go with the per-conn read
```

```
    limiting. We added a global write limit in Tor 0.1.2.5-alpha,
    but never per-conn write limits.
  - New consensus params "bwconnrate" and "bwconnburst" to let us
    rate-limit client connections as they enter the network. It’s
    controlled in the consensus so we can turn it on and off for
    experiments. It’s starting out off. Based on proposal 163.

o Major features (controllers):
  - Export GeoIP information on bridge usage to controllers even if we
    have not yet been running for 24 hours. Now Vidalia bridge operators
    can get more accurate and immediate feedback about their
    contributions to the network.
  - Add an __OwningControllerProcess configuration option and a
    TAKEOWNERSHIP control-port command. Now a Tor controller can ensure
    that when it exits, Tor will shut down. Implements feature 3049.

o Major features (directory authorities):
  - Directory authorities now create, vote on, and serve multiple
    parallel formats of directory data as part of their voting process.
    Partially implements Proposal 162: "Publish the consensus in
    multiple flavors".
  - Directory authorities now agree on and publish small summaries
    of router information that clients can use in place of regular
    server descriptors. This transition will allow Tor 0.2.3 clients
    to use far less bandwidth for downloading information about the
    network. Begins the implementation of Proposal 158: "Clients
    download consensus + microdescriptors".
  - The directory voting system is now extensible to use multiple hash
    algorithms for signatures and resource selection. Newer formats
    are signed with SHA256, with a possibility for moving to a better
    hash algorithm in the future.
  - Directory authorities can now vote on arbitary integer values as
    part of the consensus process. This is designed to help set
    network-wide parameters. Implements proposal 167.

o Major features and bugfixes (node selection):
  - Revise and reconcile the meaning of the ExitNodes, EntryNodes,
    ExcludeEntryNodes, ExcludeExitNodes, ExcludeNodes, and Strict*Nodes
    options. Previously, we had been ambiguous in describing what
    counted as an "exit" node, and what operations exactly "StrictNodes
    0" would permit. This created confusion when people saw nodes built
    through unexpected circuits, and made it hard to tell real bugs from
    surprises. Now the intended behavior is:
      . "Exit", in the context of ExitNodes and ExcludeExitNodes, means
        a node that delivers user traffic outside the Tor network.
      . "Entry", in the context of EntryNodes, means a node used as the
        first hop of a multihop circuit. It doesn't include direct
        connections to directory servers.
      . "ExcludeNodes" applies to all nodes.
      . "StrictNodes" changes the behavior of ExcludeNodes only. When
        StrictNodes is set, Tor should avoid all nodes listed in
        ExcludeNodes, even when it will make user requests fail. When
        StrictNodes is *not* set, then Tor should follow ExcludeNodes
        whenever it can, except when it must use an excluded node to
        perform self-tests, connect to a hidden service, provide a
        hidden service, fulfill a .exit request, upload directory
        information, or fetch directory information.
  Collectively, the changes to implement the behavior fix bug 1090.
  - If EntryNodes, ExitNodes, ExcludeNodes, or ExcludeExitNodes
    change during a config reload, mark and discard all our origin
    circuits. This fix should address edge cases where we change the
    config options and but then choose a circuit that we created before
    the change.
  - Make EntryNodes config option much more aggressive even when
    StrictNodes is not set. Before it would prepend your requested
    entrynodes to your list of guard nodes, but feel free to use others
    after that. Now it chooses only from your EntryNodes if any of
    those are available, and only falls back to others if a) they're
    all down and b) StrictNodes is not set.
  - Now we refresh your entry guards from EntryNodes at each consensus
    fetch -- rather than just at startup and then they slowly rot as
    the network changes.
  - Add support for the country code "{??}" in torrc options like
    ExcludeNodes, to indicate all routers of unknown country. Closes
    bug 1094.
  - ExcludeNodes now takes precedence over EntryNodes and ExitNodes:
    if a node is listed in both, it's treated as excluded.
  - ExcludeNodes now applies to directory nodes -- as a preference if
    StrictNodes is 0, or an absolute requirement if StrictNodes is 1.
    Don't exclude all the directory authorities and set StrictNodes to 1
    unless you really want your Tor to break.
  - ExcludeNodes and ExcludeExitNodes now override exit enclaving.
  - ExcludeExitNodes now overrides .exit requests.
  - We don't use bridges listed in ExcludeNodes.
  - When StrictNodes is 1:
     . We now apply ExcludeNodes to hidden service introduction points
       and to rendezvous points selected by hidden service users. This
       can make your hidden service less reliable: use it with caution!
     . If we have used ExcludeNodes on ourself, do not try relay
       reachability self-tests.
```

```
      . If we have excluded all the directory authorities, we will not
        even try to upload our descriptor if we're a relay.
      . Do not honor .exit requests to an excluded node.
    - When the set of permitted nodes changes, we now remove any
      mappings
      introduced via TrackExitHosts to now-excluded nodes. Bugfix on
      0.1.0.1-rc.
    - We never cannibalize a circuit that had excluded nodes on it, even
      if StrictNodes is 0. Bugfix on 0.1.0.1-rc.
    - Improve log messages related to excluded nodes.

  o Major features (misc):
    - Numerous changes, bugfixes, and workarounds from Nathan Freitas
      to help Tor build correctly for Android phones.
    - The options SocksPort, ControlPort, and so on now all accept a
      value "auto" that opens a socket on an OS-selected port. A
      new ControlPortWriteToFile option tells Tor to write its
      actual control port or ports to a chosen file. If the option
      ControlPortFileGroupReadable is set, the file is created as
      group-readable. Now users can run two Tor clients on the same
      system without needing to manually mess with parameters. Resolves
      part of ticket 3076.
    - Tor now supports tunneling all of its outgoing connections over
      a SOCKS proxy, using the SOCKS4Proxy and/or SOCKS5Proxy
      configuration options. Code by Christopher Davis.

  o Code security improvements:
    - Replace all potentially sensitive memory comparison operations
      with versions whose runtime does not depend on the data being
      compared. This will help resist a class of attacks where an
      adversary can use variations in timing information to learn
      sensitive data. Fix for one case of bug 3122. (Safe memcmp
      implementation by Robert Ransom based partially on code by DJB.)
    - Enable Address Space Layout Randomization (ASLR) and Data
      Execution
      Prevention (DEP) by default on Windows to make it harder for
      attackers to exploit vulnerabilities. Patch from John Brooks.
    - New "--enable-gcc-hardening" ./configure flag (off by default) to
      turn on gcc compile time hardening options. It ensures that signed
      ints have defined behavior (-fwrapv), enables -D_FORTIFY_SOURCE=2
      (requiring -O2), adds stack smashing protection with canaries
      (-fstack-protector-all), turns on ASLR protection if supported by
      the kernel (-fPIE, -pie), and adds additional security
      related warnings. Verified to work on Mac OS X and Debian Lenny.
    - New "--enable-linker-hardening" ./configure flag (off by default)
      to turn on ELF specific hardening features (relro, now). This does
      not work with Mac OS X or any other non-ELF binary format.
    - Always search the Windows system directory for system DLLs, and
      nowhere else. Bugfix on 0.1.1.23; fixes bug 1954.
    - New DisableAllSwap option. If set to 1, Tor will attempt to lock
      all
      current and future memory pages via mlockall(). On supported
      platforms (modern Linux and probably BSD but not Windows or OS X),
      this should effectively disable any and all attempts to page out
      memory. This option requires that you start your Tor as root --
      if you use DisableAllSwap, please consider using the User option
      to properly reduce the privileges of your Tor.

  o Major bugfixes (crashes):
    - Fix crash bug on platforms where gmtime and localtime can return
      NULL. Windows 7 users were running into this one. Fixes part of bug
      2077. Bugfix on all versions of Tor. Found by boboper.
    - Introduce minimum/maximum values that clients will believe
      from the consensus. Now we'll have a better chance to avoid crashes
      or worse when a consensus param has a weird value.
    - Fix a rare crash bug that could occur when a client was configured
      with a large number of bridges. Fixes bug 2629; bugfix on
      0.2.1.2-alpha. Bugfix by trac user "shitlei".
    - Do not crash when our configuration file becomes unreadable, for
      example due to a permissions change, between when we start up
      and when a controller calls SAVECONF. Fixes bug 3135; bugfix
      on 0.0.9pre6.
    - If we're in the pathological case where there's no exit bandwidth
      but there is non-exit bandwidth, or no guard bandwidth but there
      is non-guard bandwidth, don't crash during path selection. Bugfix
      on 0.2.0.3-alpha.
    - Fix a crash bug when trying to initialize the evdns module in
      Libevent 2. Bugfix on 0.2.1.16-rc.

  o Major bugfixes (stability):
    - Fix an assert in parsing router descriptors containing IPv6
      addresses. This one took down the directory authorities when
      somebody tried some experimental code. Bugfix on 0.2.1.3-alpha.
    - Fix an uncommon assertion failure when running with DNSPort under
      heavy load. Fixes bug 2933; bugfix on 0.2.0.1-alpha.
    - Treat an unset $HOME like an empty $HOME rather than triggering an
      assert. Bugfix on 0.0.8pre1; fixes bug 1522.
    - More gracefully handle corrupt state files, removing asserts
      in favor of saving a backup and resetting state.
    - Instead of giving an assertion failure on an internal mismatch
```

```
    on estimated freelist size, just log a BUG warning and try later.
    Mitigates but does not fix bug 1125.
  - Fix an assert that got triggered when using the TestingTorNetwork
    configuration option and then issuing a GETINFO config-text control
    command. Fixes bug 2250; bugfix on 0.2.1.2-alpha.
  - If the cached cert file is unparseable, warn but don't exit.

o Privacy fixes (relays/bridges):
  - Don't list Windows capabilities in relay descriptors. We never
    made use of them, and maybe it's a bad idea to publish them. Bugfix
    on 0.1.1.8-alpha.
  - If the Nickname configuration option isn't given, Tor would pick a
    nickname based on the local hostname as the nickname for a relay.
    Because nicknames are not very important in today's Tor and the
    "Unnamed" nickname has been implemented, this is now problematic
    behavior: It leaks information about the hostname without being
    useful at all. Fixes bug 2979; bugfix on 0.1.2.2-alpha, which
    introduced the Unnamed nickname. Reported by tagnaq.
  - Maintain separate TLS contexts and certificates for incoming and
    outgoing connections in bridge relays. Previously we would use the
    same TLS contexts and certs for incoming and outgoing connections.
    Bugfix on 0.2.0.3-alpha; addresses bug 988.
  - Maintain separate identity keys for incoming and outgoing TLS
    contexts in bridge relays. Previously we would use the same
    identity keys for incoming and outgoing TLS contexts. Bugfix on
    0.2.0.3-alpha; addresses the other half of bug 988.
  - Make the bridge directory authority refuse to answer directory
    requests for "all descriptors". It used to include bridge
    descriptors in its answer, which was a major information leak.
    Found by "piebeer". Bugfix on 0.2.0.3-alpha.

o Privacy fixes (clients):
  - When receiving a hidden service descriptor, check that it is for
    the hidden service we wanted. Previously, Tor would store any
    hidden service descriptors that a directory gave it, whether it
    wanted them or not. This wouldn't have let an attacker impersonate
    a hidden service, but it did let directories pre-seed a client
    with descriptors that it didn't want. Bugfix on 0.0.6.
  - Start the process of disabling ".exit" address notation, since it
    can be used for a variety of esoteric application-level attacks
    on users. To reenable it, set "AllowDotExit 1" in your torrc. Fix
    on 0.0.9rc5.
  - Reject attempts at the client side to open connections to private
    IP addresses (like 127.0.0.1, 10.0.0.1, and so on) with
    a randomly chosen exit node. Attempts to do so are always
```

```
      ill-defined, generally prevented by exit policies, and usually
      in error. This will also help to detect loops in transparent
      proxy configurations. You can disable this feature by setting
      "ClientRejectInternalAddresses 0" in your torrc.
    - Log a notice when we get a new control connection. Now it's easier
      for security-conscious users to recognize when a local application
      is knocking on their controller door. Suggested by bug 1196.

  o Privacy fixes (newnym):
    - Avoid linkability based on cached hidden service descriptors:
      forget all hidden service descriptors cached as a client when
      processing a SIGNAL NEWNYM command. Fixes bug 3000; bugfix on 0.0.6.
    - On SIGHUP, do not clear out all TrackHostExits mappings, client
      DNS cache entries, and virtual address mappings: that's what
      NEWNYM is for. Fixes bug 1345; bugfix on 0.1.0.1-rc.
    - Don't attach new streams to old rendezvous circuits after SIGNAL
      NEWNYM. Previously, we would keep using an existing rendezvous
      circuit if it remained open (i.e. if it were kept open by a
      long-lived stream, or if a new stream were attached to it before
      Tor could notice that it was old and no longer in use). Bugfix on
      0.1.1.15-rc; fixes bug 3375.

  o Major bugfixes (relay bandwidth accounting):
    - Fix a bug that could break accounting on 64-bit systems with large
      time_t values, making them hibernate for impossibly long intervals.
      Fixes bug 2146. Bugfix on 0.0.9pre6; fix by boboper.
    - Fix a bug in bandwidth accounting that could make us use twice
      the intended bandwidth when our interval start changes due to
      daylight saving time. Now we tolerate skew in stored vs computed
      interval starts: if the start of the period changes by no more than
      50% of the period's duration, we remember bytes that we transferred
      in the old period. Fixes bug 1511; bugfix on 0.0.9pre5.

  o Major bugfixes (bridges):
    - Bridges now use "reject *:*" as their default exit policy. Bugfix
      on 0.2.0.3-alpha. Fixes bug 1113.
    - If you configure your bridge with a known identity fingerprint,
      and the bridge authority is unreachable (as it is in at least
      one country now), fall back to directly requesting the descriptor
      from the bridge. Finishes the feature started in 0.2.0.10-alpha;
      closes bug 1138.
    - Fix a bug where bridge users who configure the non-canonical
      address of a bridge automatically switch to its canonical
      address. If a bridge listens at more than one address, it
      should be able to advertise those addresses independently and
      any non-blocked addresses should continue to work. Bugfix on Tor
      0.2.0.3-alpha. Fixes bug 2510.
    - If you configure Tor to use bridge A, and then quit and
      configure Tor to use bridge B instead (or if you change Tor
      to use bridge B via the controller), it would happily continue
      to use bridge A if it's still reachable. While this behavior is
      a feature if your goal is connectivity, in some scenarios it's a
      dangerous bug. Bugfix on Tor 0.2.0.1-alpha; fixes bug 2511.
    - When the controller configures a new bridge, don't wait 10 to 60
      seconds before trying to fetch its descriptor. Bugfix on
      0.2.0.3-alpha; fixes bug 3198 (suggested by 2355).

  o Major bugfixes (directory authorities):
    - Many relays have been falling out of the consensus lately because
      not enough authorities know about their descriptor for them to get
      a majority of votes. When we deprecated the v2 directory protocol,
      we got rid of the only way that v3 authorities can hear from each
      other about other descriptors. Now authorities examine every v3
      vote for new descriptors, and fetch them from that authority.
      Bugfix on 0.2.1.23.
    - Authorities could be tricked into giving out the Exit flag to
      relays that didn't allow exiting to any ports. This bug could screw
      with load balancing and stats. Bugfix on 0.1.1.6-alpha; fixes bug
      1238. Bug discovered by Martin Kowalczyk.
    - If all authorities restart at once right before a consensus vote,
      nobody will vote about "Running", and clients will get a consensus
      with no usable relays. Instead, authorities refuse to build a
      consensus if this happens. Bugfix on 0.2.0.10-alpha; fixes bug 1066.

  o Major bugfixes (stream-level fairness):
    - When receiving a circuit-level SENDME for a blocked circuit, try
      to package cells fairly from all the streams that had previously
      been blocked on that circuit. Previously, we had started with the
      oldest stream, and allowed each stream to potentially exhaust
      the circuit's package window. This gave older streams on any
      given circuit priority over newer ones. Fixes bug 1937. Detected
      originally by Camilo Viecco. This bug was introduced before the
      first Tor release, in svn commit r152: it is the new winner of
      the longest-lived bug prize.
    - Fix a stream fairness bug that would cause newer streams on a given
      circuit to get preference when reading bytes from the origin or
      destination. Fixes bug 2210. Fix by Mashael AlSabah. This bug was
      introduced before the first Tor release, in svn revision r152.
    - When the exit relay got a circuit-level sendme cell, it started
      reading on the exit streams, even if had 500 cells queued in the
```

circuit queue already, so the circuit queue just grew and grew in some cases. We fix this by not re-enabling reading on receipt of a sendme cell when the cell queue is blocked. Fixes bug 1653. Bugfix on 0.2.0.1-alpha. Detected by Mashael AlSabah. Original patch by "yetonetime".

- Newly created streams were allowed to read cells onto circuits, even if the circuit's cell queue was blocked and waiting to drain. This created potential unfairness, as older streams would be blocked, but newer streams would gladly fill the queue completely. We add code to detect this situation and prevent any stream from getting more than one free cell. Bugfix on 0.2.0.1-alpha. Partially fixes bug 1298.

o Major bugfixes (hidden services):

- Apply circuit timeouts to opened hidden-service-related circuits based on the correct start time. Previously, we would apply the circuit build timeout based on time since the circuit's creation; it was supposed to be applied based on time since the circuit entered its current state. Bugfix on 0.0.6; fixes part of bug 1297.
- Improve hidden service robustness: When we find that we have extended a hidden service's introduction circuit to a relay not listed as an introduction point in the HS descriptor we currently have, retry with an introduction point from the current descriptor. Previously we would just give up. Fixes bugs 1024 and 1930; bugfix on 0.2.0.10-alpha.
- Directory authorities now use data collected from their own uptime observations when choosing whether to assign the HSDir flag to relays, instead of trusting the uptime value the relay reports in its descriptor. This change helps prevent an attack where a small set of nodes with frequently-changing identity keys can blackhole a hidden service. (Only authorities need upgrade; others will be fine once they do.) Bugfix on 0.2.0.10-alpha; fixes bug 2709.
- Stop assigning the HSDir flag to relays that disable their DirPort (and thus will refuse to answer directory requests). This fix should dramatically improve the reachability of hidden services: hidden services and hidden service clients pick six HSDir relays to store and retrieve the hidden service descriptor, and currently about half of the HSDir relays will refuse to work. Bugfix on 0.2.0.10-alpha; fixes part of bug 1693.

o Major bugfixes (misc):

- Clients now stop trying to use an exit node associated with a given destination by TrackHostExits if they fail to reach that exit node.
  Fixes bug 2999. Bugfix on 0.2.0.20-rc.

- Fix a regression that caused Tor to rebind its ports if it receives SIGHUP while hibernating. Bugfix in 0.1.1.6-alpha; closes bug 919.
- Remove an extra pair of quotation marks around the error message in control-port STATUS_GENERAL BUG events. Bugfix on 0.1.2.6-alpha; fixes bug 3732.

o Minor features (relays):

- Ensure that no empty [dirreq-](read|write)-history lines are added to an extrainfo document. Implements ticket 2497.
- When bandwidth accounting is enabled, be more generous with how much bandwidth we'll use up before entering "soft hibernation". Previously, we'd refuse new connections and circuits once we'd used up 95% of our allotment. Now, we use up 95% of our allotment, AND make sure that we have no more than 500MB (or 3 hours of expected traffic, whichever is lower) remaining before we enter soft hibernation.
- Relays now log the reason for publishing a new relay descriptor, so we have a better chance of hunting down instances of bug 1810. Resolves ticket 3252.
- Log a little more clearly about the times at which we're no longer accepting new connections (e.g. due to hibernating). Resolves bug 2181.
- When AllowSingleHopExits is set, print a warning to explain to the relay operator why most clients are avoiding her relay.
- Send END_STREAM_REASON_NOROUTE in response to EHOSTUNREACH errors. Clients before 0.2.1.27 didn't handle NOROUTE correctly, but such clients are already deprecated because of security bugs.

o Minor features (network statistics):

- Directory mirrors that set "DirReqStatistics 1" write statistics about directory requests to disk every 24 hours. As compared to the "--enable-geoip-stats" ./configure flag in 0.2.1.x, there are a few improvements: 1) stats are written to disk exactly every 24 hours; 2) estimated shares of v2 and v3 requests are determined as mean values, not at the end of a measurement period; 3) unresolved requests are listed with country code '??'; 4) directories also measure download times.
- Exit nodes that set "ExitPortStatistics 1" write statistics on the number of exit streams and transferred bytes per port to disk every 24 hours.
- Relays that set "CellStatistics 1" write statistics on how long cells spend in their circuit queues to disk every 24 hours.
- Entry nodes that set "EntryStatistics 1" write statistics on the rough number and origins of connecting clients to disk every 24 hours.

```
    - Relays that write any of the above statistics to disk and set
      "ExtraInfoStatistics 1" include the past 24 hours of statistics in
      their extra-info documents. Implements proposal 166.

  o Minor features (GeoIP and statistics):
    - Provide a log message stating which geoip file we're parsing
      instead of just stating that we're parsing the geoip file.
      Implements ticket 2432.
    - Make sure every relay writes a state file at least every 12 hours.
      Previously, a relay could go for weeks without writing its state
      file, and on a crash could lose its bandwidth history, capacity
      estimates, client country statistics, and so on. Addresses bug 3012.
    - Relays report the number of bytes spent on answering directory
      requests in extra-info descriptors similar to {read,write}-history.
      Implements enhancement 1790.
    - Report only the top 10 ports in exit-port stats in order not to
      exceed the maximum extra-info descriptor length of 50 KB.
      Implements task 2196.
    - If writing the state file to disk fails, wait up to an hour before
      retrying again, rather than trying again each second. Fixes bug
      2346; bugfix on Tor 0.1.1.3-alpha.
    - Delay geoip stats collection by bridges for 6 hours, not 2 hours,
      when we switch from being a public relay to a bridge. Otherwise
      there will still be clients that see the relay in their consensus,
      and the stats will end up wrong. Bugfix on 0.2.1.15-rc; fixes
      bug 932.
    - Update to the August 2 2011 Maxmind GeoLite Country database.

  o Minor features (clients):
    - When expiring circuits, use microsecond timers rather than
      one-second timers. This can avoid an unpleasant situation where a
      circuit is launched near the end of one second and expired right
      near the beginning of the next, and prevent fluctuations in circuit
      timeout values.
    - If we've configured EntryNodes and our network goes away and/or all
      our entrynodes get marked down, optimistically retry them all when
      a new socks application request appears. Fixes bug 1882.
    - Always perform router selections using weighted relay bandwidth,
      even if we don't need a high capacity circuit at the time. Non-fast
      circuits now only differ from fast ones in that they can use relays
      not marked with the Fast flag. This "feature" could turn out to be
      a horrible bug; we should investigate more before it goes into a
      stable release.
    - When we run out of directory information such that we can't build
      circuits, but then get enough that we can build circuits, log when
      we actually construct a circuit, so the user has a better chance of
      knowing what's going on. Fixes bug 1362.
    - Log SSL state transitions at debug level during handshake, and
      include SSL states in error messages. This may help debug future
      SSL handshake issues.

  o Minor features (directory authorities):
    - When a router changes IP address or port, authorities now launch
      a new reachability test for it. Implements ticket 1899.
    - Directory authorities now reject relays running any versions of
      Tor between 0.2.1.3-alpha and 0.2.1.18 inclusive; they have
      known bugs that keep RELAY_EARLY cells from working on rendezvous
      circuits. Followup to fix for bug 2081.
    - Directory authorities now reject relays running any version of Tor
      older than 0.2.0.26-rc. That version is the earliest that fetches
      current directory information correctly. Fixes bug 2156.
    - Directory authorities now do an immediate reachability check as soon
      as they hear about a new relay. This change should slightly reduce
      the time between setting up a relay and getting listed as running
      in the consensus. It should also improve the time between setting
      up a bridge and seeing use by bridge users.
    - Directory authorities no longer launch a TLS connection to every
      relay as they startup. Now that we have 2k+ descriptors cached,
      the resulting network hiccup is becoming a burden. Besides,
      authorities already avoid voting about Running for the first half
      hour of their uptime.
    - Directory authorities now log the source of a rejected POSTed v3
      networkstatus vote, so we can track failures better.
    - Backport code from 0.2.3.x that allows directory authorities to
      clean their microdescriptor caches. Needed to resolve bug 2230.

  o Minor features (hidden services):
    - Use computed circuit-build timeouts to decide when to launch
      parallel introduction circuits for hidden services. (Previously,
      we would retry after 15 seconds.)
    - Don't allow v0 hidden service authorities to act as clients.
      Required by fix for bug 3000.
    - Ignore SIGNAL NEWNYM commands on relay-only Tor instances. Required
      by fix for bug 3000.
    - Make hidden services work better in private Tor networks by not
      requiring any uptime to join the hidden service descriptor
      DHT. Implements ticket 2088.
    - Log (at info level) when purging pieces of hidden-service-client
      state because of SIGNAL NEWNYM.
```

```
o Minor features (controller interface):
  - New "GETINFO net/listeners/(type)" controller command to return
    a list of addresses and ports that are bound for listeners for a
    given connection type. This is useful when the user has configured
    "SocksPort auto" and the controller needs to know which port got
    chosen. Resolves another part of ticket 3076.
  - Have the controller interface give a more useful message than
    "Internal Error" in response to failed GETINFO requests.
  - Add a TIMEOUT_RATE keyword to the BUILDTIMEOUT_SET control port
    event, to give information on the current rate of circuit timeouts
    over our stored history.
  - The ’EXTENDCIRCUIT’ control port command can now be used with
    a circ id of 0 and no path. This feature will cause Tor to build
    a new ’fast’ general purpose circuit using its own path selection
    algorithms.
  - Added a BUILDTIMEOUT_SET controller event to describe changes
    to the circuit build timeout.
  - New controller command "getinfo config-text". It returns the
    contents that Tor would write if you send it a SAVECONF command,
    so the controller can write the file to disk itself.

o Minor features (controller protocol):
  - Add a new ControlSocketsGroupWritable configuration option: when
    it is turned on, ControlSockets are group-writeable by the default
    group of the current user. Patch by Jérémy Bobbio; implements
    ticket 2972.
  - Tor now refuses to create a ControlSocket in a directory that is
    world-readable (or group-readable if ControlSocketsGroupWritable
    is 0). This is necessary because some operating systems do not
    enforce permissions on an AF_UNIX sockets. Permissions on the
    directory holding the socket, however, seems to work everywhere.
  - Warn when CookieAuthFileGroupReadable is set but CookieAuthFile is
    not. This would lead to a cookie that is still not group readable.
    Closes bug 1843. Suggested by katmagic.
  - Future-proof the controller protocol a bit by ignoring keyword
    arguments we do not recognize.

o Minor features (more useful logging):
  - Revise most log messages that refer to nodes by nickname to
    instead use the "$key=nickname at address" format. This should be
    more useful, especially since nicknames are less and less likely
    to be unique. Resolves ticket 3045.
  - When an HTTPS proxy reports "403 Forbidden", we now explain
    what it means rather than calling it an unexpected status code.
    Closes bug 2503. Patch from Michael Yakubovich.
  - Rate-limit a warning about failures to download v2 networkstatus
    documents. Resolves part of bug 1352.
  - Rate-limit the "your application is giving Tor only an IP address"
    warning. Addresses bug 2000; bugfix on 0.0.8pre2.
  - Rate-limit "Failed to hand off onionskin" warnings.
  - When logging a rate-limited warning, we now mention how many messages
    got suppressed since the last warning.
  - Make the formerly ugly "2 unknown, 7 missing key, 0 good, 0 bad,
    2 no signature, 4 required" messages about consensus signatures
    easier to read, and make sure they get logged at the same severity
    as the messages explaining which keys are which. Fixes bug 1290.
  - Don’t warn when we have a consensus that we can’t verify because
    of missing certificates, unless those certificates are ones
    that we have been trying and failing to download. Fixes bug 1145.

o Minor features (log domains):
  - Add documentation for configuring logging at different severities in
    different log domains. We’ve had this feature since 0.2.1.1-alpha,
    but for some reason it never made it into the manpage. Fixes
    bug 2215.
  - Make it simpler to specify "All log domains except for A and B".
    Previously you needed to say "[*,˜A,˜B]". Now you can just say
    "[˜A,˜B]".
  - Add a "LogMessageDomains 1" option to include the domains of log
    messages along with the messages. Without this, there’s no way
    to use log domains without reading the source or doing a lot
    of guessing.
  - Add a new "Handshake" log domain for activities that happen
    during the TLS handshake.

o Minor features (build process):
  - Make compilation with clang possible when using
    "--enable-gcc-warnings" by removing two warning options that clang
    hasn’t implemented yet and by fixing a few warnings. Resolves
    ticket 2696.
  - Detect platforms that brokenly use a signed size_t, and refuse to
    build there. Found and analyzed by doorss and rransom.
  - Fix a bunch of compile warnings revealed by mingw with gcc 4.5.
    Resolves bug 2314.
  - Add support for statically linking zlib by specifying
    "--enable-static-zlib", to go with our support for statically
    linking openssl and libevent. Resolves bug 1358.
  - Instead of adding the svn revision to the Tor version string,
    report the git commit (when we’re building from a git checkout).
  - Rename the "log.h" header to "torlog.h" so as to conflict with
```

```
      fewer system headers.
    - New --digests command-line switch to output the digests of the
      source files Tor was built with.
    - Generate our manpage and HTML documentation using Asciidoc. This
      change should make it easier to maintain the documentation, and
      produce nicer HTML. The build process fails if asciidoc cannot
      be found and building with asciidoc isn't disabled (via the
      "--disable-asciidoc" argument to ./configure. Skipping the manpage
      speeds up the build considerably.

  o Minor features (options / torrc):
    - Warn when the same option is provided more than once in a torrc
      file, on the command line, or in a single SETCONF statement, and
      the option is one that only accepts a single line. Closes bug 1384.
    - Warn when the user configures two HiddenServiceDir lines that point
      to the same directory. Bugfix on 0.0.6 (the version introducing
      HiddenServiceDir); fixes bug 3289.
    - Add new "perconnbwrate" and "perconnbwburst" consensus params to
      do individual connection-level rate limiting of clients. The torrc
      config options with the same names trump the consensus params, if
      both are present. Replaces the old "bwconnrate" and "bwconnburst"
      consensus params which were broken from 0.2.2.7-alpha through
      0.2.2.14-alpha. Closes bug 1947.
    - New config option "WarnUnsafeSocks 0" disables the warning that
      occurs whenever Tor receives a socks handshake using a version of
      the socks protocol that can only provide an IP address (rather
      than a hostname). Setups that do DNS locally over Tor are fine,
      and we shouldn't spam the logs in that case.
    - New config option "CircuitStreamTimeout" to override our internal
      timeout schedule for how many seconds until we detach a stream from a
      circuit and try a new circuit. If your network is particularly slow,
      you might want to set this to a number like 60.
    - New options for SafeLogging to allow scrubbing only log messages
      generated while acting as a relay. Specify "SafeLogging relay" if
      you want to ensure that only messages known to originate from
      client use of the Tor process will be logged unsafely.
    - Time and memory units in the configuration file can now be set to
      fractional units. For example, "2.5 GB" is now a valid value for
      AccountingMax.
    - Support line continuations in the torrc config file. If a line
      ends with a single backslash character, the newline is ignored, and
      the configuration value is treated as continuing on the next line.
      Resolves bug 1929.

  o Minor features (unit tests):
    - Revise our unit tests to use the "tinytest" framework, so we
      can run tests in their own processes, have smarter setup/teardown
      code, and so on. The unit test code has moved to its own
      subdirectory, and has been split into multiple modules.
    - Add a unit test for cross-platform directory-listing code.
    - Add some forgotten return value checks during unit tests. Found
      by coverity.
    - Use GetTempDir to find the proper temporary directory location on
      Windows when generating temporary files for the unit tests. Patch
      by Gisle Vanem.

  o Minor features (misc):
    - The "torify" script now uses torsocks where available.
    - Make Libevent log messages get delivered to controllers later,
      and not from inside the Libevent log handler. This prevents unsafe
      reentrant Libevent calls while still letting the log messages
      get through.
    - Certain Tor clients (such as those behind check.torproject.org)
      may want to fetch the consensus in an extra early manner. To enable
      this a user may now set FetchDirInfoExtraEarly to 1. This also depends
      on setting FetchDirInfoEarly to 1. Previous behavior will stay the
      same as only certain clients who must have this information sooner
      should set this option.
    - Expand homedirs passed to tor-checkkey. This should silence a
      coverity complaint about passing a user-supplied string into
      open() without checking it.
    - Make sure to disable DirPort if running as a bridge. DirPorts aren’t
      used on bridges, and it makes bridge scanning somewhat easier.
    - Create the /var/run/tor directory on startup on OpenSUSE if it is
      not already created. Patch from Andreas Stieger. Fixes bug 2573.

  o Minor bugfixes (relays):
    - When a relay decides that its DNS is too broken for it to serve
      as an exit server, it advertised itself as a non-exit, but
      continued to act as an exit. This could create accidental
      partitioning opportunities for users. Instead, if a relay is
      going to advertise reject *:* as its exit policy, it should
      really act with exit policy "reject *:*". Fixes bug 2366.
      Bugfix on Tor 0.1.2.5-alpha. Bugfix by user "postman" on trac.
    - Publish a router descriptor even if generating an extra-info
      descriptor fails. Previously we would not publish a router
      descriptor without an extra-info descriptor; this can cause fast
      exit relays collecting exit-port statistics to drop from the
      consensus. Bugfix on 0.1.2.9-rc; fixes bug 2195.
    - When we’re trying to guess whether we know our IP address as
```

```
      a relay, we would log various ways that we failed to guess
      our address, but never log that we ended up guessing it
      successfully. Now add a log line to help confused and anxious
      relay operators. Bugfix on 0.1.2.1-alpha; fixes bug 1534.
    - For bandwidth accounting, calculate our expected bandwidth rate
      based on the time during which we were active and not in
      soft-hibernation during the last interval. Previously, we were
      also considering the time spent in soft-hibernation. If this
      was a long time, we would wind up underestimating our bandwidth
      by a lot, and skewing our wakeup time towards the start of the
      accounting interval. Fixes bug 1789. Bugfix on 0.0.9pre5.
    - Demote a confusing TLS warning that relay operators might get when
      someone tries to talk to their ORPort. It is not the operator's
      fault, nor can they do anything about it. Fixes bug 1364; bugfix
      on 0.2.0.14-alpha.
    - Change "Application request when we're believed to be offline."
      notice to "Application request when we haven't used client
      functionality lately.", to clarify that it's not an error. Bugfix
      on 0.0.9.3; fixes bug 1222.

  o Minor bugfixes (bridges):
    - When a client starts or stops using bridges, never use a circuit
      that was built before the configuration change. This behavior could
      put at risk a user who uses bridges to ensure that her traffic only
      goes to the chosen addresses. Bugfix on 0.2.0.3-alpha; fixes bug 3200.
    - Do not reset the bridge descriptor download status every time we
      re-parse our configuration or get a configuration change. Fixes
      bug 3019; bugfix on 0.2.0.3-alpha.
    - Users couldn't configure a regular relay to be their bridge. It
      didn't work because when Tor fetched the bridge descriptor, it found
      that it already had it, and didn't realize that the purpose of the
      descriptor had changed. Now we replace routers with a purpose other
      than bridge with bridge descriptors when fetching them. Bugfix on
      0.1.1.9-alpha. Fixes bug 1776.
    - In the special case where you configure a public exit relay as
      your bridge, Tor would be willing to use that exit relay as the last
      hop in your circuit as well. Now we fail that circuit instead.
      Bugfix on 0.2.0.12-alpha. Fixes bug 2403. Reported by "piebeer".

  o Minor bugfixes (clients):
    - We now ask the other side of a stream (the client or the exit)
      for more data on that stream when the amount of queued data on
      that stream dips low enough. Previously, we wouldn't ask the
      other side for more data until either it sent us more data (which
      it wasn't supposed to do if it had exhausted its window!) or we
      had completely flushed all our queued data. This flow control fix
      should improve throughput. Fixes bug 2756; bugfix on the earliest
      released versions of Tor (svn commit r152).
    - When a client finds that an origin circuit has run out of 16-bit
      stream IDs, we now mark it as unusable for new streams.  Previously,
      we would try to close the entire circuit. Bugfix on 0.0.6.
    - Make it explicit that we don't cannibalize one-hop circuits. This
      happens in the wild, but doesn't turn out to be a problem because
      we fortunately don't use those circuits. Many thanks to outofwords
      for the initial analysis and to swissknife who confirmed that
      two-hop circuits are actually created.
    - Resolve an edge case in path weighting that could make us misweight
      our relay selection. Fixes bug 1203; bugfix on 0.0.8rc1.
    - Make the DNSPort option work with libevent 2.x. Don't alter the
      behaviour for libevent 1.x. Fixes bug 1143. Found by SwissTorExit.

  o Minor bugfixes (directory authorities):
    - Make directory authorities more accurate at recording when
      relays that have failed several reachability tests became
      unreachable, so we can provide more accuracy at assigning Stable,
      Guard, HSDir, etc flags. Bugfix on 0.2.0.6-alpha. Resolves bug 2716.
    - Directory authorities are now more robust to hops back in time
      when calculating router stability. Previously, if a run of uptime
      or downtime appeared to be negative, the calculation could give
      incorrect results. Bugfix on 0.2.0.6-alpha; noticed when fixing
      bug 1035.
    - Directory authorities will now attempt to download consensuses
      if their own efforts to make a live consensus have failed. This
      change means authorities that restart will fetch a valid
      consensus, and it means authorities that didn't agree with the
      current consensus will still fetch and serve it if it has enough
      signatures. Bugfix on 0.2.0.9-alpha; fixes bug 1300.
    - Never vote for a server as "Running" if we have a descriptor for
      it claiming to be hibernating, and that descriptor was published
      more recently than our last contact with the server. Bugfix on
      0.2.0.3-alpha; fixes bug 911.
    - Directory authorities no longer change their opinion of, or vote
      on, whether a router is Running, unless they have themselves been
      online long enough to have some idea. Bugfix on 0.2.0.6-alpha.
      Fixes bug 1023.

  o Minor bugfixes (hidden services):
    - Log malformed requests for rendezvous descriptors as protocol
      warnings, not warnings. Also, use a more informative log message
      in case someone sees it at log level warning without prior
```

```
      info-level messages. Fixes bug 2748; bugfix on 0.2.0.10-alpha.
    - Accept hidden service descriptors if we think we might be a hidden
      service directory, regardless of what our consensus says. This
      helps robustness, since clients and hidden services can sometimes
      have a more up-to-date view of the network consensus than we do,
      and if they think that the directory authorities list us a HSDir,
      we might actually be one. Related to bug 2732; bugfix on
      0.2.0.10-alpha.
    - Correct the warning displayed when a rendezvous descriptor exceeds
      the maximum size. Fixes bug 2750; bugfix on 0.2.1.5-alpha. Found by
      John Brooks.
    - Clients and hidden services now use HSDir-flagged relays for
      hidden service descriptor downloads and uploads even if the relays
      have no DirPort set and the client has disabled TunnelDirConns. This
      will eventually allow us to give the HSDir flag to relays with no
      DirPort. Fixes bug 2722; bugfix on 0.2.1.6-alpha.
    - Only limit the lengths of single HS descriptors, even when multiple
      HS descriptors are published to an HSDir relay in a single POST
      operation. Fixes bug 2948; bugfix on 0.2.1.5-alpha. Found by hsdir.

  o Minor bugfixes (controllers):
    - Allow GETINFO fingerprint to return a fingerprint even when
      we have not yet built a router descriptor. Fixes bug 3577;
      bugfix on 0.2.0.1-alpha.
    - Send a SUCCEEDED stream event to the controller when a reverse
      resolve succeeded. Fixes bug 3536; bugfix on 0.0.8pre1. Issue
      discovered by katmagic.
    - Remove a trailing asterisk from "exit-policy/default" in the
      output of the control port command "GETINFO info/names". Bugfix
      on 0.1.2.5-alpha.
    - Make the SIGNAL DUMP controller command work on FreeBSD. Fixes bug
      2917. Bugfix on 0.1.1.1-alpha.
    - When we restart our relay, we might get a successful connection
      from the outside before we’ve started our reachability tests,
      triggering a warning: "ORPort found reachable, but I have no
      routerinfo yet. Failing to inform controller of success." This
      bug was harmless unless Tor is running under a controller
      like Vidalia, in which case the controller would never get a
      REACHABILITY_SUCCEEDED status event. Bugfix on 0.1.2.6-alpha;
      fixes bug 1172.
    - When a controller changes TrackHostExits, remove mappings for
      hosts that should no longer have their exits tracked. Bugfix on
      0.1.0.1-rc.
    - When a controller changes VirtualAddrNetwork, remove any mappings
      for hosts that were automapped to the old network. Bugfix on
      0.1.1.19-rc.
    - When a controller changes one of the AutomapHosts* options, remove
      any mappings for hosts that should no longer be automapped. Bugfix
      on 0.2.0.1-alpha.
    - Fix an off-by-one error in calculating some controller command
      argument lengths. Fortunately, this mistake is harmless since
      the controller code does redundant NUL termination too. Found by
      boboper. Bugfix on 0.1.1.1-alpha.
    - Fix a bug in the controller interface where "GETINFO ns/asdaskljkl"
      would return "551 Internal error" rather than "552 Unrecognized key
      ns/asdaskljkl". Bugfix on 0.1.2.3-alpha.
    - Don't spam the controller with events when we have no file
      descriptors available. Bugfix on 0.2.1.5-alpha. (Rate-limiting
      for log messages was already solved from bug 748.)
    - Emit a GUARD DROPPED controller event for a case we missed.
    - Ensure DNS requests launched by "RESOLVE" commands from the
      controller respect the __LeaveStreamsUnattached setconf options.
      The same goes for requests launched via DNSPort or transparent
      proxying. Bugfix on 0.2.0.1-alpha; fixes bug 1525.

  o Minor bugfixes (config options):
    - Tor used to limit HttpProxyAuthenticator values to 48 characters.
      Change the limit to 512 characters by removing base64 newlines.
      Fixes bug 2752. Fix by Michael Yakubovich.
    - Complain if PublishServerDescriptor is given multiple arguments that
      include 0 or 1. This configuration will be rejected in the future.
      Bugfix on 0.2.0.1-alpha; closes bug 1107.
    - Disallow BridgeRelay 1 and ORPort 0 at once in the configuration.
      Bugfix on 0.2.0.13-alpha; closes bug 928.

  o Minor bugfixes (log subsystem fixes):
    - When unable to format an address as a string, report its value
      as "???" rather than reusing the last formatted address. Bugfix
      on 0.2.1.5-alpha.
    - Be more consistent in our treatment of file system paths. "~" should
      get expanded to the user's home directory in the Log config option.
      Fixes bug 2971; bugfix on 0.2.0.1-alpha, which introduced the feature
      for the -f and --DataDirectory options.

  o Minor bugfixes (memory management):
    - Don't stack-allocate the list of supplementary GIDs when we're
      about to log them. Stack-allocating NGROUPS_MAX gid_t elements
      could take up to 256K, which is way too much stack. Found by
      Coverity; CID #450. Bugfix on 0.2.1.7-alpha.
    - Save a couple bytes in memory allocation every time we escape
      certain characters in a string. Patch from Florian Zumbiehl.

  o Minor bugfixes (protocol correctness):
    - When checking for 1024-bit keys, check for 1024 bits, not 128
      bytes. This allows Tor to correctly discard keys of length 1017
      through 1023. Bugfix on 0.0.9pre5.
    - Require that introduction point keys and onion handshake keys
      have a public exponent of 65537. Starts to fix bug 3207; bugfix
      on 0.2.0.10-alpha.
    - Handle SOCKS messages longer than 128 bytes long correctly, rather
      than waiting forever for them to finish. Fixes bug 2330; bugfix
      on 0.2.0.16-alpha. Found by doorss.
    - Never relay a cell for a circuit we have already destroyed.
      Between marking a circuit as closeable and finally closing it,
      it may have been possible for a few queued cells to get relayed,
      even though they would have been immediately dropped by the next
      OR in the circuit. Fixes bug 1184; bugfix on 0.2.0.1-alpha.
    - Never queue a cell for a circuit that's already been marked
      for close.
    - Fix a spec conformance issue: the network-status-version token
      must be the first token in a v3 consensus or vote. Discovered by
      "parakeep". Bugfix on 0.2.0.3-alpha.
    - A networkstatus vote must contain exactly one signature. Spec
      conformance issue. Bugfix on 0.2.0.3-alpha.
    - When asked about a DNS record type we don't support via a
      client DNSPort, reply with NOTIMPL rather than an empty
      reply. Patch by intrigeri. Fixes bug 3369; bugfix on 2.0.1-alpha.
    - Make more fields in the controller protocol case-insensitive,
      since control-spec.txt said they were.

  o Minor bugfixes (log messages):
    - Fix a log message that said "bits" while displaying a value in
      bytes. Found by wanoskarnet. Fixes bug 3318; bugfix on
      0.2.0.1-alpha.
    - Downgrade "no current certificates known for authority" message
      from Notice to Info. Fixes bug 2899; bugfix on 0.2.0.10-alpha.
    - Correctly describe errors that occur when generating a TLS object.
      Previously we would attribute them to a failure while generating a
      TLS context. Patch by Robert Ransom. Bugfix on 0.1.0.4-rc; fixes
      bug 1994.
    - Fix an instance where a Tor directory mirror might accidentally
      log the IP address of a misbehaving Tor client. Bugfix on
      0.1.0.1-rc.
    - Stop logging at severity 'warn' when some other Tor client tries
      to establish a circuit with us using weak DH keys. It's a protocol
```

```
      violation, but that doesn't mean ordinary users need to hear about
      it. Fixes the bug part of bug 1114. Bugfix on 0.1.0.13.
    - If your relay can't keep up with the number of incoming create
      cells, it would log one warning per failure into your logs. Limit
      warnings to 1 per minute. Bugfix on 0.0.2pre10; fixes bug 1042.

  o Minor bugfixes (build fixes):
    - Fix warnings from GCC 4.6's "-Wunused-but-set-variable" option.
    - When warning about missing zlib development packages during
      compile, give the correct package names. Bugfix on 0.2.0.1-alpha.
    - Fix warnings that newer versions of autoconf produce during
      ./autogen.sh. These warnings appear to be harmless in our case,
      but they were extremely verbose. Fixes bug 2020.
    - Squash a compile warning on OpenBSD. Reported by Tas; fixes
      bug 1848.

  o Minor bugfixes (portability):
    - Write several files in text mode, on OSes that distinguish text
      mode from binary mode (namely, Windows). These files are:
      'buffer-stats', 'dirreq-stats', and 'entry-stats' on relays
      that collect those statistics; 'client_keys' and 'hostname' for
      hidden services that use authentication; and (in the tor-gencert
      utility) newly generated identity and signing keys. Previously,
      we wouldn't specify text mode or binary mode, leading to an
      assertion failure. Fixes bug 3607. Bugfix on 0.2.1.1-alpha (when
      the DirRecordUsageByCountry option which would have triggered
      the assertion failure was added), although this assertion failure
      would have occurred in tor-gencert on Windows in 0.2.0.1-alpha.
    - Selectively disable deprecation warnings on OS X because Lion
      started deprecating the shipped copy of openssl. Fixes bug 3643.
    - Use a wide type to hold sockets when built for 64-bit Windows.
      Fixes bug 3270.
    - Fix an issue that prevented static linking of libevent on
      some platforms (notably Linux). Fixes bug 2698; bugfix on 0.2.1.23,
      where we introduced the "--with-static-libevent" configure option.
    - Fix a bug with our locking implementation on Windows that couldn't
      correctly detect when a file was already locked. Fixes bug 2504,
      bugfix on 0.2.1.6-alpha.
    - Build correctly on OSX with zlib 1.2.4 and higher with all
      warnings enabled.
    - Fix IPv6-related connect() failures on some platforms (BSD, OS X).
      Bugfix on 0.2.0.3-alpha; fixes first part of bug 2660. Patch by
      "piebeer".

  o Minor bugfixes (code correctness):
```

- Always NUL-terminate the sun_path field of a sockaddr_un before passing it to the kernel. (Not a security issue: kernels are smart enough to reject bad sockaddr_uns.) Found by Coverity; CID #428. Bugfix on Tor 0.2.0.3-alpha.
- Make connection_printf_to_buf()'s behaviour sane. Its callers expect it to emit a CRLF iff the format string ends with CRLF; it actually emitted a CRLF iff (a) the format string ended with CRLF or (b) the resulting string was over 1023 characters long or (c) the format string did not end with CRLF *and* the resulting string was 1021 characters long or longer. Bugfix on 0.1.1.9-alpha; fixes part of bug 3407.
- Make send_control_event_impl()'s behaviour sane. Its callers expect it to always emit a CRLF at the end of the string; it might have emitted extra control characters as well. Bugfix on 0.1.1.9-alpha; fixes another part of bug 3407.
- Make crypto_rand_int() check the value of its input correctly. Previously, it accepted values up to UINT_MAX, but could return a negative number if given a value above INT_MAX+1. Found by George Kadianakis. Fixes bug 3306; bugfix on 0.2.2pre14.
- Fix a potential null-pointer dereference while computing a consensus. Bugfix on tor-0.2.0.3-alpha, found with the help of clang's analyzer.
- If we fail to compute the identity digest of a v3 legacy keypair, warn, and don't use a buffer-full of junk instead. Bugfix on 0.2.1.1-alpha; fixes bug 3106.
- Resolve an untriggerable issue in smartlist_string_num_isin(), where if the function had ever in the future been used to check for the presence of a too-large number, it would have given an incorrect result. (Fortunately, we only used it for 16-bit values.) Fixes bug 3175; bugfix on 0.1.0.1-rc.
- Be more careful about reporting the correct error from a failed connect() system call. Under some circumstances, it was possible to look at an incorrect value for errno when sending the end reason. Bugfix on 0.1.0.1-rc.
- Correctly handle an "impossible" overflow cases in connection byte counting, where we write or read more than 4GB on an edge connection in a single second. Bugfix on 0.1.2.8-beta.
- Avoid a double mark-for-free warning when failing to attach a transparent proxy connection. Bugfix on 0.1.2.1-alpha. Fixes bug 2279.
- Correctly detect failure to allocate an OpenSSL BIO. Fixes bug 2378; found by "cypherpunks". This bug was introduced before the first Tor release, in svn commit r110.
- Fix a bug in bandwidth history state parsing that could have been triggered if a future version of Tor ever changed the timing

```
    granularity at which bandwidth history is measured. Bugfix on
    Tor 0.1.1.11-alpha.
  - Add assertions to check for overflow in arguments to base32_encode()
    and base32_decode(); fix a signed-unsigned comparison there too. These
    bugs are not actually reachable in Tor, but it’s good to prevent
    future errors too. Found by doorss.
  - Avoid a bogus overlapped memcpy in tor_addr_copy(). Reported by
    "memcpyfail".
  - Set target port in get_interface_address6() correctly. Bugfix
    on 0.1.1.4-alpha and 0.2.0.3-alpha; fixes second part of bug 2660.
  - Fix an impossible-to-actually-trigger buffer overflow in relay
    descriptor generation. Bugfix on 0.1.0.15.
  - Fix numerous small code-flaws found by Coverity Scan Rung 3.

o Minor bugfixes (code improvements):
  - After we free an internal connection structure, overwrite it
    with a different memory value than we use for overwriting a freed
    internal circuit structure. Should help with debugging. Suggested
    by bug 1055.
  - If OpenSSL fails to make a duplicate of a private or public key, log
    an error message and try to exit cleanly. May help with debugging
    if bug 1209 ever remanifests.
  - Some options used different conventions for uppercasing of acronyms
    when comparing manpage and source. Fix those in favor of the
    manpage, as it makes sense to capitalize acronyms.
  - Take a first step towards making or.h smaller by splitting out
    function definitions for all source files in src/or/. Leave
    structures and defines in or.h for now.
  - Remove a few dead assignments during router parsing. Found by
    coverity.
  - Don’t use 1-bit wide signed bit fields. Found by coverity.
  - Avoid signed/unsigned comparisons by making SIZE_T_CEILING unsigned.
    None of the cases where we did this before were wrong, but by making
    this change we avoid warnings. Fixes bug 2475; bugfix on 0.2.1.28.
  - The memarea code now uses a sentinel value at the end of each area
    to make sure nothing writes beyond the end of an area. This might
    help debug some conceivable causes of bug 930.
  - Always treat failure to allocate an RSA key as an unrecoverable
    allocation error.
  - Add some more defensive programming for architectures that can’t
    handle unaligned integer accesses. We don’t know of any actual bugs
    right now, but that’s the best time to fix them. Fixes bug 1943.

o Minor bugfixes (misc):
  - Fix a rare bug in rend_fn unit tests: we would fail a test when
```

```
      a randomly generated port is 0. Diagnosed by Matt Edman. Bugfix
      on 0.2.0.10-alpha; fixes bug 1808.
    - Where available, use Libevent 2.0's periodic timers so that our
      once-per-second cleanup code gets called even more closely to
      once per second than it would otherwise. Fixes bug 943.
    - Ignore OutboundBindAddress when connecting to localhost.
      Connections to localhost need to come _from_ localhost, or else local
      servers (like DNS and outgoing HTTP/SOCKS proxies) will often refuse
      to listen.
    - Update our OpenSSL 0.9.8l fix so that it works with OpenSSL 0.9.8m
      too.
    - If any of the v3 certs we download are unparseable, we should
      actually notice the failure so we don't retry indefinitely. Bugfix
      on 0.2.0.x; reported by "rotator".
    - When Tor fails to parse a descriptor of any kind, dump it to disk.
      Might help diagnosing bug 1051.
    - Make our 'torify' script more portable; if we have only one of
      'torsocks' or 'tsocks' installed, don't complain to the user;
      and explain our warning about tsocks better.
    - Fix some urls in the exit notice file and make it XHTML1.1 strict
      compliant. Based on a patch from Christian Kujau.

  o Documentation changes:
    - Modernize the doxygen configuration file slightly. Fixes bug 2707.
    - Resolve all doxygen warnings except those for missing documentation.
      Fixes bug 2705.
    - Add doxygen documentation for more functions, fields, and types.
    - Convert the HACKING file to asciidoc, and add a few new sections
      to it, explaining how we use Git, how we make changelogs, and
      what should go in a patch.
    - Document the default socks host and port (127.0.0.1:9050) for
      tor-resolve.
    - Removed some unnecessary files from the source distribution. The
      AUTHORS file has now been merged into the people page on the
      website. The roadmaps and design doc can now be found in the
      projects directory in svn.

  o Deprecated and removed features (config):
    - Remove the torrc.complete file. It hasn't been kept up to date
      and users will have better luck checking out the manpage.
    - Remove the HSAuthorityRecordStats option that version 0 hidden
      service authorities could use to track statistics of overall v0
      hidden service usage.
    - Remove the obsolete "NoPublish" option; it has been flagged
      as obsolete and has produced a warning since 0.1.1.18-rc.
    - Caches no longer download and serve v2 networkstatus documents
      unless FetchV2Networkstatus flag is set: these documents haven't
      haven't been used by clients or relays since 0.2.0.x. Resolves
      bug 3022.

  o Deprecated and removed features (controller):
    - The controller no longer accepts the old obsolete "addr-mappings/"
      or "unregistered-servers-" GETINFO values.
    - The EXTENDED_EVENTS and VERBOSE_NAMES controller features are now
      always on; using them is necessary for correct forward-compatible
      controllers.

  o Deprecated and removed features (misc):
    - Hidden services no longer publish version 0 descriptors, and
      clients do not request or use version 0 descriptors. However, the
      old hidden service authorities still accept and serve version 0
      descriptors when contacted by older hidden services/clients.
    - Remove undocumented option "-F" from tor-resolve: it hasn't done
      anything since 0.2.1.16-rc.
    - Remove everything related to building the expert bundle for OS X.
      It has confused many users, doesn't work right on OS X 10.6,
      and is hard to get rid of once installed. Resolves bug 1274.
    - Remove support for .noconnect style addresses. Nobody was using
      them, and they provided another avenue for detecting Tor users
      via application-level web tricks.
    - When we fixed bug 1038 we had to put in a restriction not to send
      RELAY_EARLY cells on rend circuits. This was necessary as long
      as relays using Tor 0.2.1.3-alpha through 0.2.1.18-alpha were
      active. Now remove this obsolete check. Resolves bug 2081.
    - Remove workaround code to handle directory responses from servers
      that had bug 539 (they would send HTTP status 503 responses _and_ send
      a body too). Since only server versions before 0.2.0.16-alpha/0.1.2.19
      were affected, there is no longer reason to keep the workaround
      in place.
    - Remove the old 'fuzzy time' logic. It was supposed to be used for
      handling calculations where we have a known amount of clock skew
      and an allowed amount of unknown skew. But we only used it in three
      places, and we never adjusted the known/unknown skew values. This
      is still something we might want to do someday, but if we do, we'll
      want to do it differently.
    - Remove the "--enable-iphone" option to ./configure. According to
      reports from Marco Bonetti, Tor builds fine without any special
      tweaking on recent iPhone SDK versions.
```

7. On August 28th, we released Torbutton 1.4.1.

Torbutton 1.4.1 has been released at:
https://www.torproject.org/torbutton/

This release features a "New Identity" menu option that clears browser state, closes tabs, and obtains a fresh Tor circuit for future requests. It also features a fix for breakage with Hotmail, and further isolates browser state and identifiers to the url bar domain (see https://blog.torproject.org/blog/improving-private-browsing-modes-do-not-track-vs-real-privacy-design).

However, the New Identity button and the Hotmail fix are only available to Tor Browser Bundle users. If you are still using Torbutton with a vanilla Mozilla Firefox, we strongly recommend you download the Tor Browser Bundle 2.2.x alphas from https://www.torproject.org/dist/torbrowser/ and report problems you experience, as we will be declaring them stable within the next release or two.

Here is the complete changelog:

* bug 523: Implement New Identity (for TBB only)
* bug 3580: Fix hotmail/live breakage (for TBB only)
* bug 3748: Disable 3rd party HTTP auth
* bug 3665: Fix several corner cases SafeCache isolation
* bug 3739: Fix https->http CORS failure for SafeCache
* bug 3414: Isolate window.name based on referrer policy
* bug 3809: Disable referer spoofing (fixes navigation issues)
* bug 3819: Fix API issue with cookie protections
* bug 3820: Fix warning w/ session store filter

8. On August 31st, updated the stable Tor Browser Bundle.

We have updated the stable Tor Browser Bundles to Firefox 6.

There are no longer any stable Tor Browser Bundles with the 3.6.x series of Firefox. We were using pre-built binaries on some platforms and owing to the recent DigiNotar debacle, we no longer felt comfortable shipping versions of Firefox that we were unable to patch. We build all Firefox 6 binaries from source, with our own set of patches, including some specific to the DigiNotar issue.

We'd originally planned to drop support for Firefox 3.6 bundles on September 10th, but this moved up the date a bit. The new Tor Browser Bundles are much more feature-rich than the previous bundles, but users may still experience unexpected behavior. Please report all bugs to https://trac.torproject.org/.

```
Windows users will see the biggest difference between the old stable
bundle and the new stable bundle. In addition to upgrading Firefox, it
includes the latest stable release of Tor 0.2.2.32,Vidalia 0.2.14 and
Torbutton 1.4.1. These three upgrades together allow you to run the Tor
Browser Bundle at the same time as a system Tor, or even multiple copies
of the Tor Browser Bundle in different directories, by dynamically
choosing available ports.

https://www.torproject.org/download

Tor Browser Bundle (2.2.32-2)

    Update Firefox to 6.0.1, with an additional patch to exclude DigiNotar
    completely
        https://gitweb.torproject.org/torbrowser.git/commit/0be3b043afa0e54d207f

        For the full saga, read:
        http://blog.mozilla.com/security/2011/08/29/fraudulent-google-com-certif...
        http://googleonlinesecurity.blogspot.com/2011/08/update-on-attempted-man...
    OS X specific: Rebuild 32-bit binaries with backwards compatibility
    options so TBB works on OSX 10.5 (closes: #3671)
    Update Libevent to 2.0.14-stable
    Update torbrowser.version string in prefs.js to have more
    information (see #3504)
    Enable internationalized bundles by adding and changing the
    general.useragent.locale pref in prefs.js
```

Design, develop, and implement enhancements that make Tor a better tool for users in censored countries.

- Fixed a bunch of bugs related to microdescriptors and Proposal 171 code. These will show up in September Tor development releases.
- Released multiple versions of Tor Browser Bundle to address issues with the Diginotar Certificate Authority crack.
  https://blog.torproject.org/blog/diginotar-debacle-and-what-you-should-do-about-it
  and
  https://blog.torproject.org/blog/diginotar-damage-disclosure
- Sebastian rewrote TorCheck and Tor Bulk Exit List (TorBEL) tools. We're going to do some more testing and then deploy the updated TorBEL, Check, and DNSel tools in October 2011.
  - Tor Check: It currently comes with a very simplistic torcheck replacement that Sebastian is embedding the new html into:

  https://trac.torproject.org/projects/tor/ticket/3749 along with a new torbutton feature for TBB upgrade checking:
  https://trac.torproject.org/projects/tor/ticket/2285

  - Bulk Exit List: We need a replacement for the current bulk exit list script, because it makes thousands of DNS requests and kills the network as well as the host that it runs on. Torbel did not have a script like this, but Sebastian wrote one. It is much faster than the old script.
  - DNS exit list service: This was the core of the old infrastructure; now it is just an addon to TorBEL. This means that even if the DNS service gets overloaded massively, check and bulk list can function independently.
  - Core TorBEL: TorBEL itself is now just a service that publishes an exit list along with a status file that tells when the list will be updated next. This means that all the other services making use of the data can simply fetch the list and parse it instead of making one request per request they receive themselves.

- Steven worked on Tor patches to support UPnP on Windows, which also will be useful for launching pluggable transport helper programs. This is now merged into Tor, along with the following patches to fix the test cases.

  These changes consist of:

  - bug 2046: teach Tor how to launch background subprocesses on Windows:
    https://trac.torproject.org/projects/tor/ticket/2046
  - bug 1983: get tor-fw-helper to build on Windows:
    https://trac.torproject.org/projects/tor/ticket/1983
  - Patch miniupnpc to build on Windows:
    https://gitweb.torproject.org/user/sjm217/miniupnpc-patches.git

  This code works for two testers, so we think it is now ready for more users. It is in Tor 0.2.3.3-alpha, but disabled by default.

- Set up a preview of the new Tor Status written by the Wesleyan students in order to get more feedback from the community. Tried to move the discussion forward how to maintain TorStatus in the future and how to merge the relevant metrics website parts into it. The relevant links are:

  https://blog.torproject.org/blog/new-pythondjango-based-torstatus
  http://www.torstatusbeta.org/

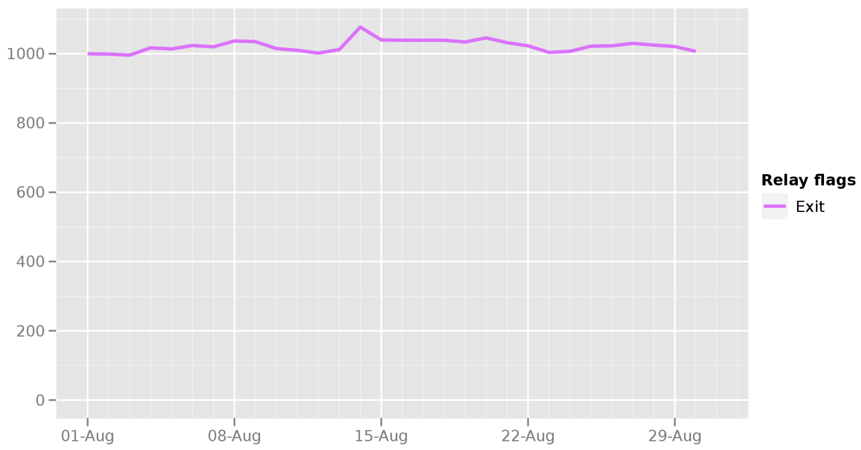
Hide Tor's network signature.

- We're working with CAIDA to get datasets to better test and tune obfsproxy. We've been working through the permissions – it looks like George's aggregated version of their dataset should be something we can release to the world.

Grow the Tor network and user base. Outreach.

Measures of the Tor Network

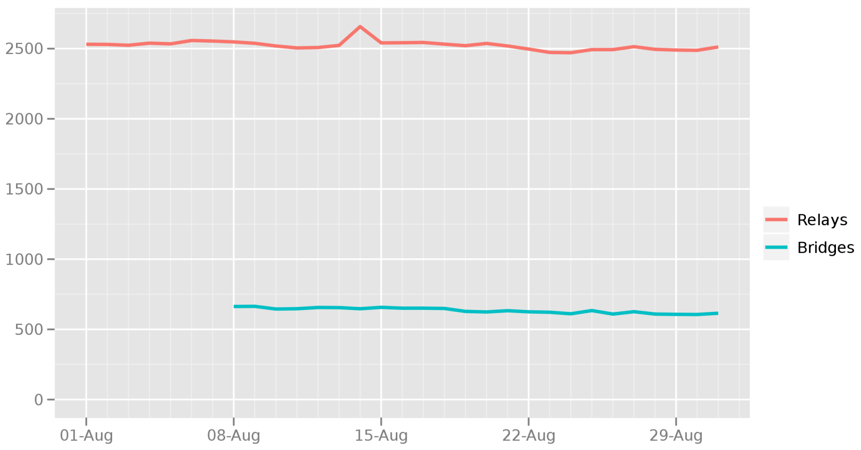
Number of relays with relay flags assigned



The Tor Project - https://metrics.torproject.org/

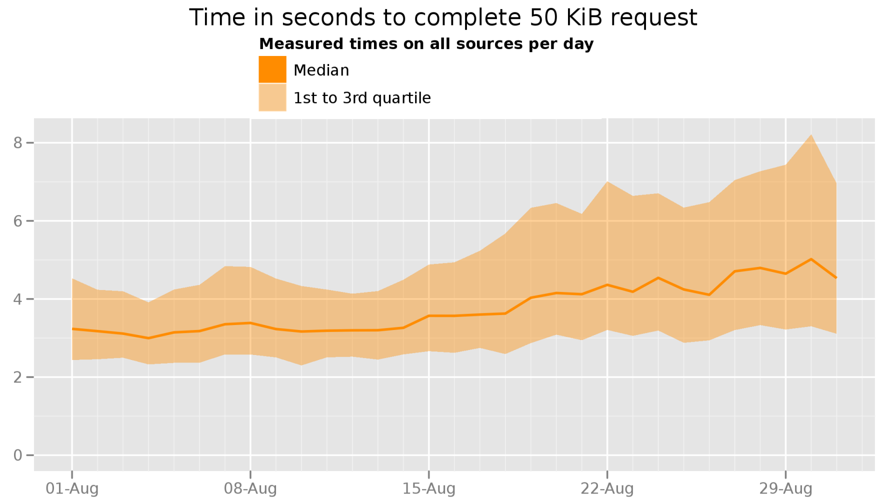
This graph shows the total quantity of exit relays in August 2011. There is a very slight reduction in relays over the month.

Number of relays



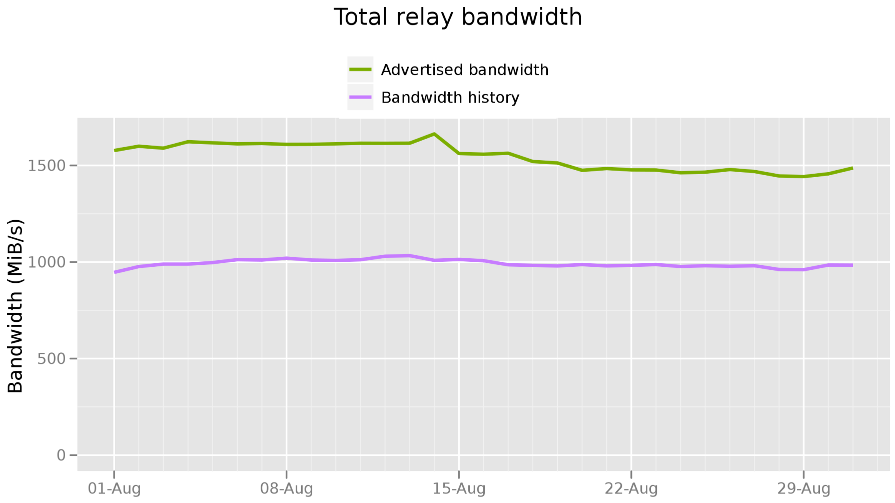
The Tor Project - https://metrics.torproject.org/

This graph shows the total quantity of relays and the total quantity of bridges in August 2011.



The Tor Project - https://metrics.torproject.org/

This graphs shows how many seconds it took to complete a 50KB download from a standard Tor client. This is an average of all measurements from servers located in Illinois, Massachusetts, and Sweden. Average latency has increased slightly to 4.5 seconds.



The Tor Project - https://metrics.torproject.org/

This graph shows the total available bandwidth available to clients and how much was actually used throughout the month. The steady-state of relays creates almost 1.5GBps (12.0 Gbps) of bandwidth available.

Outreach and Advocacy

1. The Persian News Network ran a campaign to spread Tor to their users. This campaign has resulted in over 10,000 emails delivering Farsi Tor Browser Bundle. We've seen Tor usage from

Iran increase to around 40,000 users daily. https://metrics.torproject.org/packages.html

2. Roger attended FOCI, the Workshop on Free and Open Communications on the Internet: https://db.usenix.org/events/foci11/tech/.
3. Roger attended Usenix Security Symposium, https://db.usenix.org/events/sec11/index.html.
4. Roger presented the keynote at Crypto, http://www.iacr.org/conferences/crypto2011/.
5. Worked with EFF on their blog post about IP Addresses alone do not identify criminals, https://www.eff.org/deeplinks/2011/08/why-ip-addresses-alone-dont-identify-criminals.
6. Roger posted a research question to the community, https://blog.torproject.org/blog/research-problem-better-guard-rotation-parameters.
7. Runa attended DEFCON and represented Tor at the Privacy By Design "Develop4Privacy" awards, https://www.develop4privacy.org/awards.
8. Andrew presented at the National Network to End Domestic Violence annual technical conference.
   https://blog.torproject.org/blog/ending-domestic-violence-nnedv-and-tor
9. Announced the new Tor stable via press release. https://www.torproject.org/press/2011-08-28-tor-022-stable
10. We started to gather a set of "user stories" on the blog:
    https://blog.torproject.org/blog/we-need-your-good-tor-stories

Preconfigured privacy (circumvention) bundles for USB or LiveCD.

- Due to the end of life of Firefox 4 and 5, and the DigiNotar Certificate Authority breach, we released Firefox 6 as part of Tor Browser Bundle. Details on the related Tor Browser releases and the DigiNotar issues are at https://blog.torproject.org/category/tags/ohdiginotaryoudidnt

Bridge relay and bridge authority work.

- Continued to make progress on the hardware Tor router device based on the DreamPlug. We have a skeleton graphical interface to allow users to configure it through a web browser. Integration of the arm, console relay controller continues.

Scalability, load balancing, directory overhead, efficiency.

- Started discussion of sanitizing our web server logs. Wrote a simple parser to sanitize existing logs, discussed sanitizing procedure on tor-dev. https://lists.torproject.org/pipermail/tor-dev/2011-August/002892.html

- Added data formats page to metrics website, https://metrics.torproject.org/formats.html.

- Fixed a strange out-of-memory problem on metrics by assigning 8G of RAM to a Java program instead of 4G, both of which are too much. Some of this code needs a rewrite that can handle the growing amounts of data.

- Performed experiments with Amazon EC2 cloud. See https://github.com/inf0/Tor-Cloud for progress. Code cleanup and finalization on this is scheduled for September. Tor now has an Amazon AWS account with developers assigned to it for testing the Tor Cloud concept.

Incentives work.

Nothing to report.

More reliable (e.g. split) download mechanism.

- PNN Tsunami: The PNN Tsunami brought up some usability issues of the GetTor service. Christian tried to address them, with the help of Roger. We re-wrote large parts of the help text and fixed some usability bugs on the way:
  https://trac.torproject.org/projects/tor/ticket/3379
  https://trac.torproject.org/projects/tor/ticket/3381
  https://trac.torproject.org/projects/tor/ticket/2796
  https://trac.torproject.org/projects/tor/ticket/3772
  https://trac.torproject.org/projects/tor/ticket/3829
  https://trac.torproject.org/projects/tor/ticket/3855

- Deployed a redesigned download page to help guide users to the right software for their operating system and language.

Footprints from Tor Browser Bundle.

Nothing to report.

Translation work, ultimately a browser-based approach.

- Sebastian and Runa prototyped a script that automatically fetches translations from transifex and commits them to git.

- Updated translations in German, Arabic, Greek, Spanish, Farsi, Portugese Brazilian, Russian, Vietnamese, and Simplified Chinese.