dr.kaos and *kaos.theory*

proudly present

Building an Anonym.OS

v1.0

Hardened, **O**ptimized, **T**ransportable
System for **E**ncrypting and **A**nonymizing **T**raffic

Goals of Anonym.OS

- Secure your system
- Eliminate telltale footprints
- Prevent or reduce effectiveness of fingerprinting
- Bypass restrictive filters
- Ensure confidentiality, integrity
- Simulate outside connections

Choose Your Path

- What's more important to you, encryption or anonymity?
- Are you concerned about performance?
- What OS do you feel comfortable with?
- Are you willing to invest in commercial sw?

Target Operating Systems

- Linux
  - Easy, can be accomplished today
- *BSD
  - Easy, can be accomplished today
- Mac OS X
  - Essentially just BSD
  - Apple’s customizations, however, increase chattiness
  - More complex than Linux/Free/OpenBSD

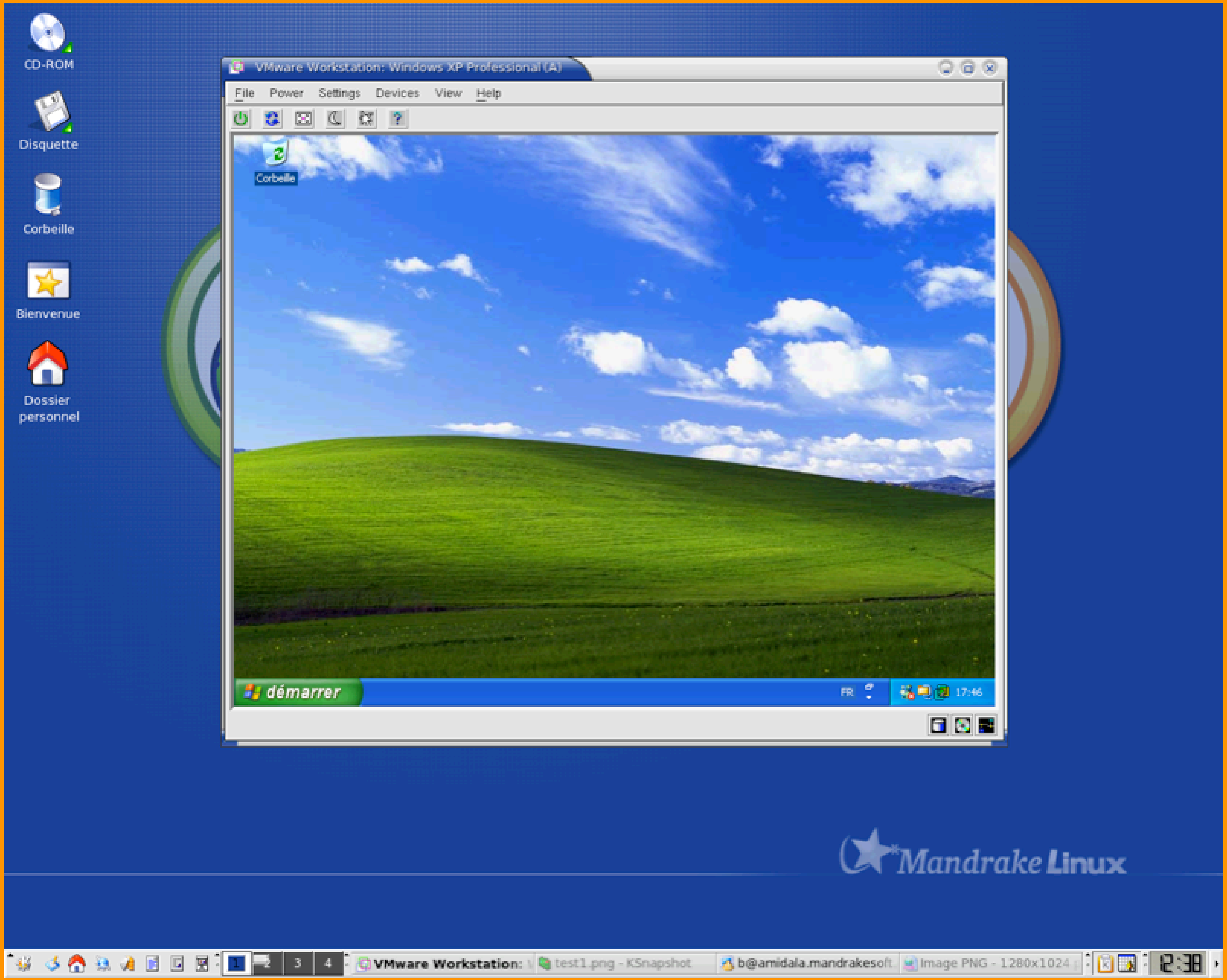
What about Windows?

- If you're using Windows, are we really supposed to believe you care about security, anonymity and encryption? ;)
- If so, we're working on a few partial solutions to accomplish this task
  - Unfortunately, it's pretty low priority for us atm
  - In the meantime, look to commercial software for a native solution (i.e. BlackIce, ZoneAlarm, etc)
- Or...

Windows via Emulation

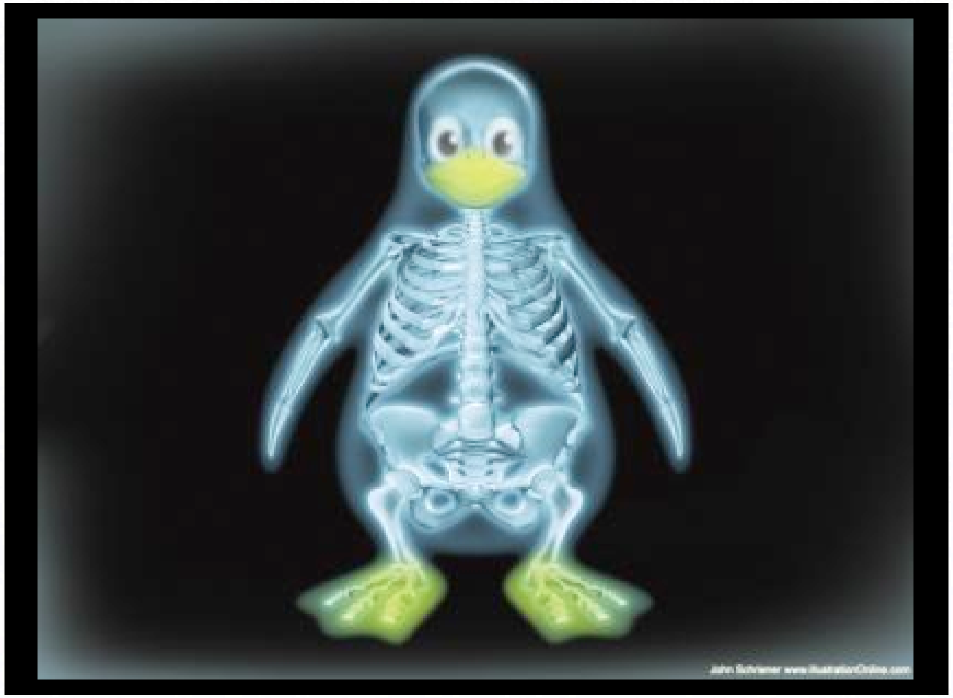
- Using VMWare, Virtual PC, or similar emulation software (i.e. Bochs), users can perform effective ingress and egress filtering with Linux, BSD or OSX as a VM host
- Unfortunately, it still may be difficult to force some Windows applications to utilize your anonymizing and encrypting proxies

Windows via Emulation



Building the Anonym.OS

1. Host Hardening
2. Strong Ingress Filtering
3. Strong Egress Filtering
4. Content-Filtering Proxies
5. Anonymizing Proxies
6. Encrypted Protocols



Alc^H^H^H Linux
Anonym.OS

Hardening Linux

- Disable unnecessary running services
  - Implement TCPwrappers / xinetd for others
- Delete unnecessary files / packages
- Implement kernel-based security patches
- *Automate hardening via Bastille

Disabling Services

- Redhat / Fedora:

```
/sbin/chkconfig service off
```

- Debian:

```
update-rc.d service remove
```

- Gentoo:

```
rc-update del service default
```

Deleting Packages

- Redhat / Fedora:

```
rpm -e package
```

- Debian:

```
apt-get remove package
```

- Gentoo:

```
emerge -C package
```

Kernel Security Patches

- Numerous patches exist for the Linux 2.4 and 2.6 kernel trees:
  - LSM
  - Grsecurity, PaX
  - LIDS
  - SELinux
  - Immunix AppArmor
- Stock distro kernels may include some of these patches by default

LIDS Overview

- No one can modify lids-protected files; files can be hidden
- No one can modify lids-protected processes; processes can be hidden
- Provides network access restrictions
- Fine-grained access control via simple ACL's
- Security alerts from the kernel
- Port scanner detection in kernel
- Supports LSM in 2.5+ kernels

LIDS Example

- First, patch kernel (LIDS is current to 2.6.11)
- Second, build ACLs for your OS, ex:

```
# Protect System Binaries
#
/sbin/lidsconf -A -o /sbin                               -j READONLY
/sbin/lidsconf -A -o /bin                                -j READONLY

# Protect System Configuration files
#
/sbin/lidsconf -A -o /etc                                -j READONLY
/sbin/lidsconf -A -o /usr/local/etc                      -j READONLY
/sbin/lidsconf -A -o /etc/shadow                         -j DENY
/sbin/lidsconf -A -o /etc/lilo.conf                      -j DENY
```

- Extensive Examples at http://www.lids.org

grsecurity / PaX

- Intelligent RBAC with minimal configuration
- Chroot hardening
- /tmp race prevention
- Pax prevents class of addr space exploits
- *Addt'l randomness in TCP/IP stack*
- Users only view their own processes
- Extensive auditing, tied back to originating IP

Bastille

- Scripts to harden *nix operating systems, including:
    - Redhat, Debian, Gentoo, Mandrake, SuSE, TurboLinux
    - Mac OS X
    - HP-sUX
- Most effective on virgin machine/install
- Very instructive approach to hardening

Packet Filtering

- Typically, administrators will configure firewalls with strong ingress filtering rules, but minimal if any egress filtering rules
- For ***Anonym.OS***, egress rules are at least as important as ingress rules, if not more so
- In Linux, we can perform both ingress and egress filtering using Netfilter / IPTables

Netfilter / IPTables

- Foundation for packet filtering, NAT, PAT and general packet mangling in 2.4 / 2.6 kernels
- Performs statefull ingress and egress filtering
- Also enables modification of other fields within IP header, for ex. TOS/ECN etc.

Filtering w/ IPTables

- `MYIP=86.75.30.9`
- `# Set default policy to drop`
- `iptables -P INPUT DROP`
- `iptables -P OUTPUT DROP`
- `# Flush all tables`
- `iptables -F`
- `iptables -F INPUT`
- `iptables -F OUTPUT`
- `# Drop all outbound packets not from us`
- `iptables -A OUTPUT -o eth0 -s ! $MYIP -j DROP`
- `# Allow specific outbound traffic, ex 9050`

-
  ```
  iptables -A OUTPUT -o eth0 -p tcp -s $MYIP \
  --dport 9050 -d 1.2.3.4 -j ACCEPT
  ```
- `# Force egress traffic through a local proxy`

-
  ```
  iptables -t nat -A PREROUTING -i eth0 \
  -p tcp --dport 80 -j REDIRECT --to-port 8118
  ```

IP Personality Patch

- Only for 2.4 kernels
- Designed to defeat basic fingerprinting, i.e. as performed by nmap
- Characteristics that can be changed:
  - TCP initial sequence number
  - TCP initial window size
  - TCP options
  - IP ID numbers

Content Filtering Proxies

- Eliminate junk and reduce bandwidth consumption
- Minimize fingerprints (user agent, etc)
- Numerous content filtering proxies exist, pick your favorite, ex:
    - Privoxy
    - RabbIT Proxy
    - WebCleaner

Privoxy

- Designed to protect privacy
  - Can modify webpage content
  - Manages cookies
  - Can control access
  - Blocks ads, banners, popups
- Easy to install, configure, use
- Binaries available for Linux, Mac, Windows

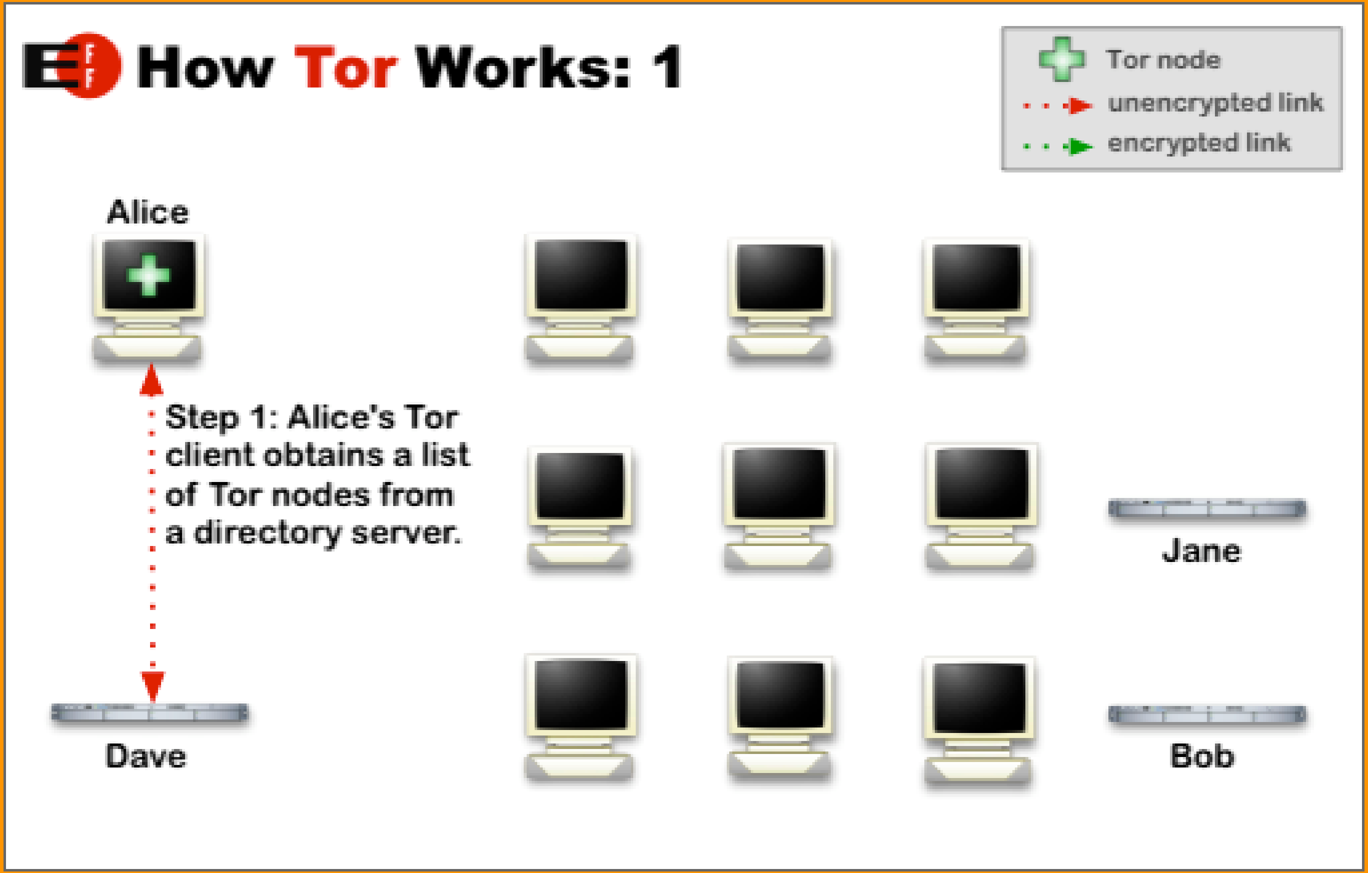
Anonymizing Proxies

- Provide basic anonymous browsing
- Some support encryption
- Numerous lists are available on the Internet
- Firefox has an extension called SwitchProxy that is designed to assist with setting up chained Anonymous Proxies

The Contenders

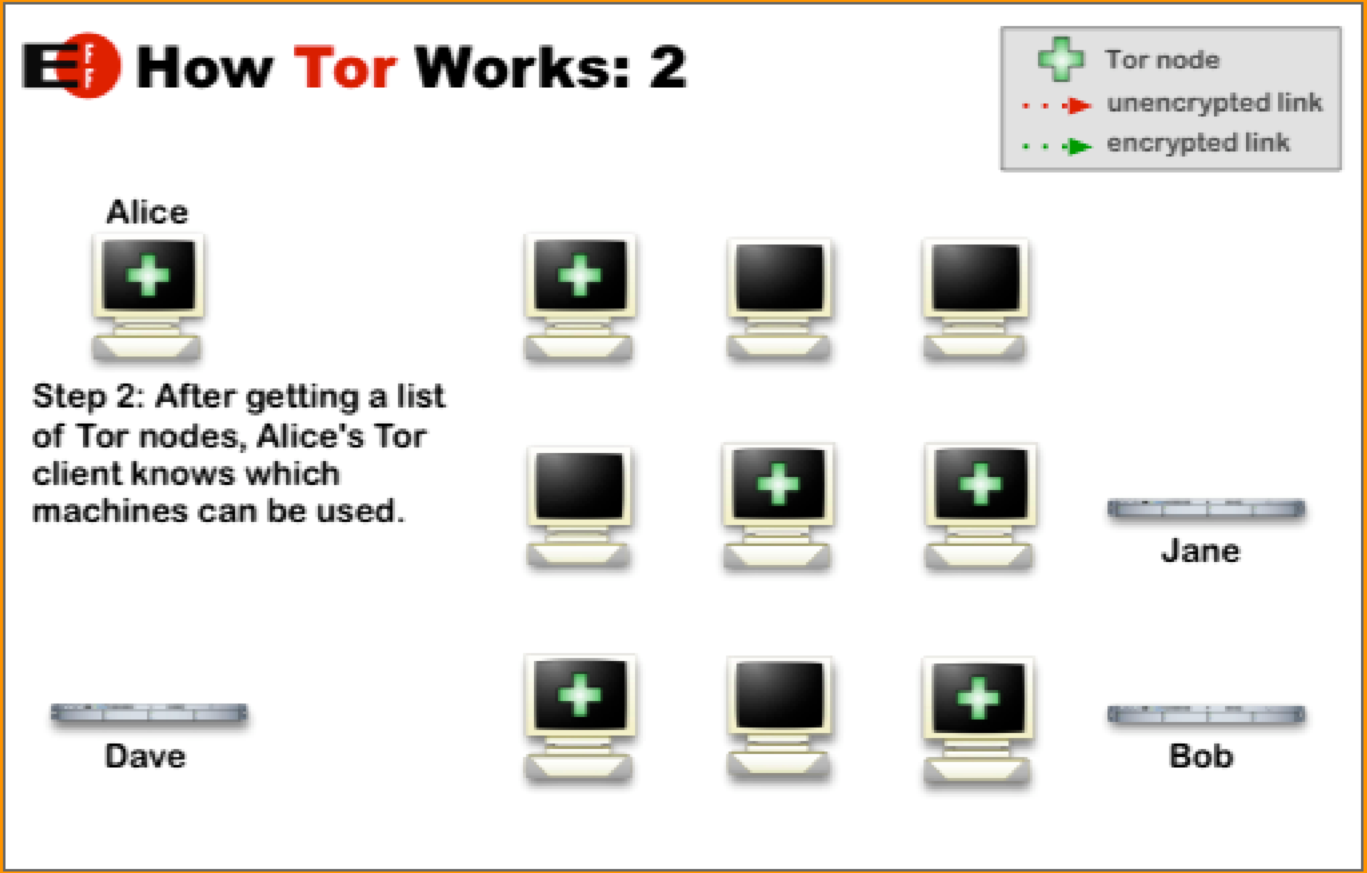
- JAP
  - Uses a single static address for all users
  - Users take encrypted detour through several intermediaries in predetermined “mix”
- Tor
  - Protection against traffic analysis
  - Hides you amongst other users in the network
  - Only works for TCP streams
  - Works with any app that supports SOCKS
  - Sponsored by the EFF :)

Tor



Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

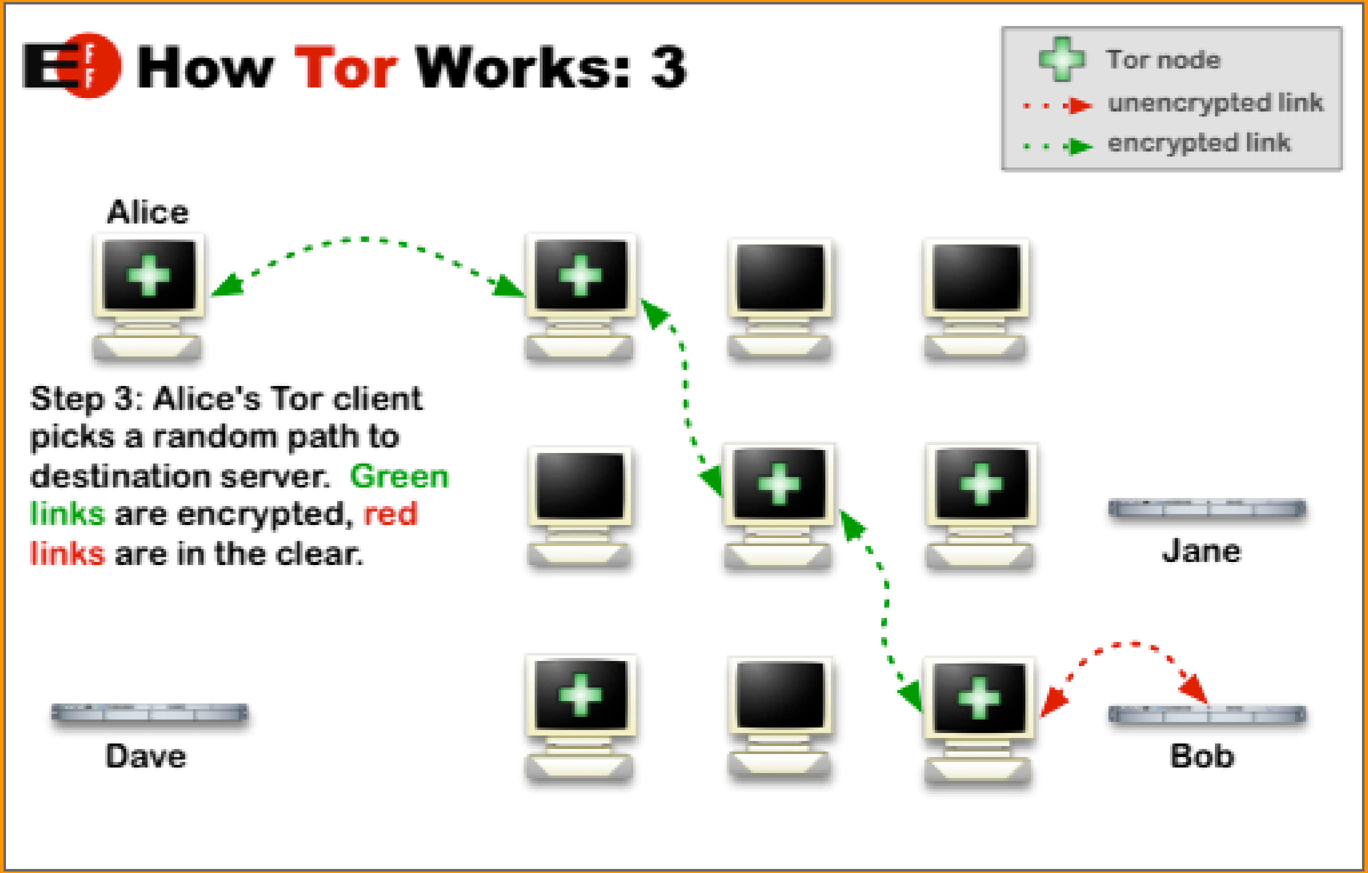
Tor



How Tor Works: 2

Step 2: After getting a list of Tor nodes, Alice's Tor client knows which machines can be used.

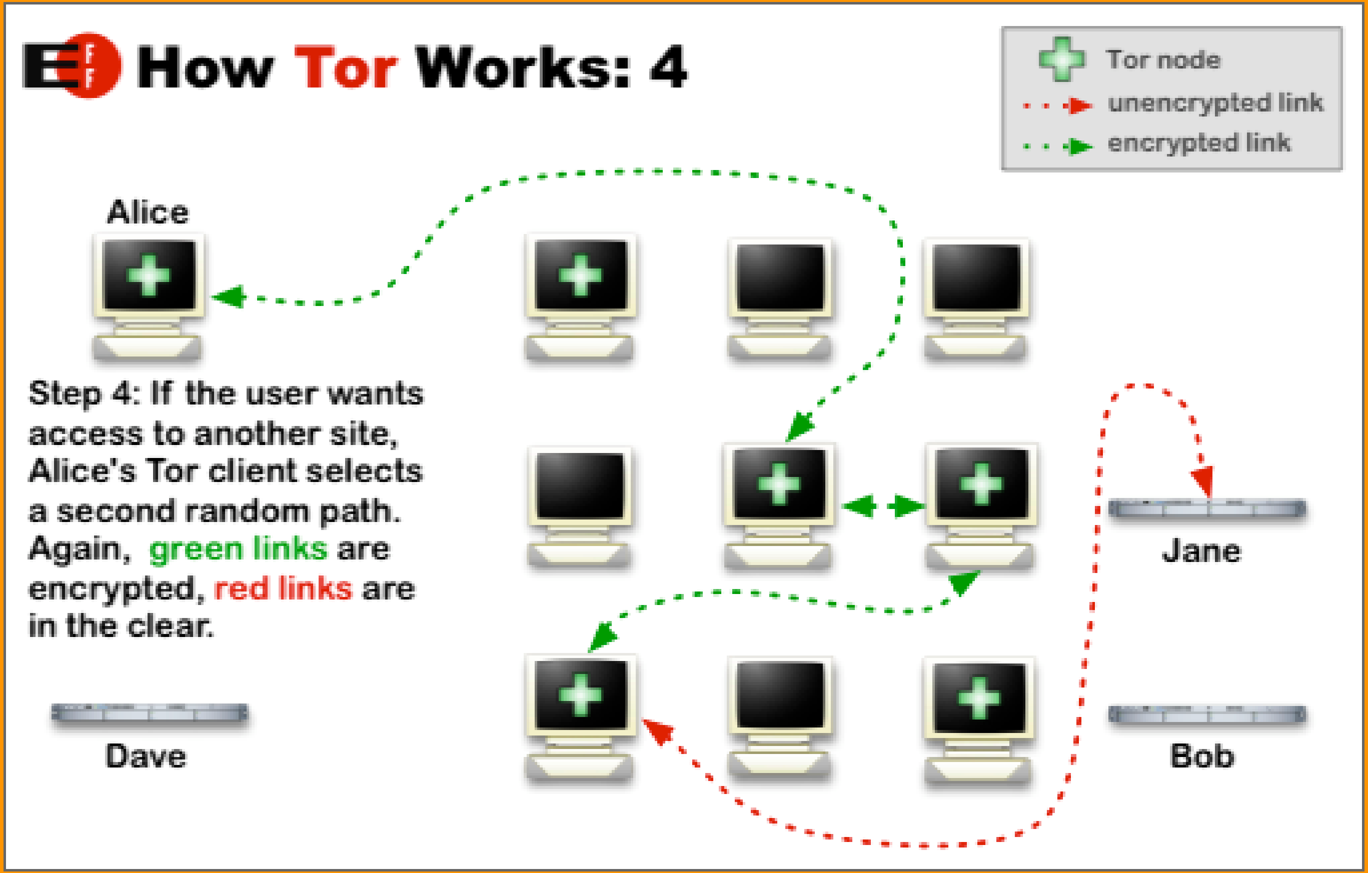
Tor



How Tor Works: 3

Step 3: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Tor



How Tor Works: 4

Step 4: If the user wants access to another site, Alice's Tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

BSD/Mac Anonym.OS

Hardening BSD/Mac

- Procedure is similar to linux, in that you will still:
  - Disable unnecessary services
    (Mac OS X, look in /Library/StartupServices)
  - Delete unneeded binaries
  - Bastille is available for Mac OS X too!
- BSD typically uses IPFW or PF for packet filtering

Mac OS X Firewalling

- Mac OS X uses IPFW, which is configured by the “Firewall” pane in Sytem Preferences app
- Commercial app *Little Snitch* can provide supplemental egress filtering for Preference Pane ingress filtering
- Firewall pane can be disabled/circumvented and custom IPFW rules used at every startup

Mac OS X Firewalling

- Create a StartupItem to start new firewalls rules
- /Library/StartupItems/Firewall/
  /Library/StartupItems/Firewall/StartupParameters.plist
  /Library/StartupItems/Firewall:

- `# First flush the firewall rules`
- `$FW -q flush`
- `# Allow all traffic from the loopback interface`
- `$FW add allow all from any to any via lo0`
- `# Deny all other traffic`
- `$FW add 65534 deny log ip from any to any`

Mac Proxies

- Squid, Privoxy, JAP and Tor are all available for Mac OS X
- Tor package from EFF site will automatically install and configure both Privoxy *and* Tor
- SquidMan GUI available for Mac, which makes easy the configuration of upstream proxies

Don’t Forget the Client

- Individual application settings may give away personally identifying information
- Browsers are probably best example:
  - Default browser configurations are typically not favorable for users concerned about anonymity/privacy
- Numerous other applications exhibit similar invasive characteristics, so watch out!
  - Auto send registration
  - Auto check for updates
  - Auto look-up CD info
  - Auto submit error/bug report

Privacy in Mozilla

- Use Mozillas JavaScript Popup filtering.
  See Mozilla -> Edit -> Preferences -> Advanced -> Scripts & Plugins
- Disable Java.
  See Mozilla -> Edit -> Preferences -> Advanced
- Don't send your real email address to FTP servers.
  See Mozilla -> Edit -> Preferences -> Advanced
- Don't accept cookies, or at least set the browser to warn you of every attempt to store cookies.
  See Mozilla -> Edit -> Preferences -> Privacy & Security -> Cookies
- Disable image animation.
  See Mozilla -> Edit -> Preferences -> Privacy & Security -> Images
- Don't save form data.
  See Mozilla -> Edit -> Preferences -> Privacy & Security -> Forms
- Don't save passwords.
  See Mozilla -> Edit -> Preferences -> Privacy & Security -> Passwords
- Don't install the Flash Plugin. It has security problems.

Close to Optimal

- Hardened OS
- No network services running
- Ingress DROP ALL
  Egress REDIRECT SPECIFIC to proxy, DROP REST
- Local proxy chain: Privoxy -> Tor
  (want caching? try Squid -> Privoxy -> Tor)
- Carefully-tweaked client applications
- Using all encrypted protocols
  (HTTPS, IMAPS, POP3S, SMTP+TLS, etc)

Limitations

- Often, performance sucks :)
- Some protocols cannot be easily anonymized
- Some applications may not work properly through proxies
- Encryption and anonymity are sometimes at cross-purposes
- Only a sniffer can tell you how effective your Anonym.OS is!

The Future

- v2 of this presentation (no typos ;)
- Walkthrough documents, with specifics, for Gentoo and Mac OS X
- ***kaos.theory Anonym.OS LiveCD***
- More on Windows

Thanks

- ***beth*** and ***digunix*** for their ideas, insight and contributions to this project
- the rest of ***kaos.theory*** for helping me turn this into more than just a side project

References

- Paste links here :)