Speaking Anonymously In Public
A Hacker's Guide

Pope Alan Bradley I
(The Space Pope)
Sirius B, Canis Majoris, Interstellar Space
{abradley at fastmail.fm}

Abstract

This paper is meant to provide instructions on how to execute an anonymous speech at a technical conference. These techniques have been used successfully by Alan Bradley and the Kevin Flynns to present “Tron: He Fights For The User”[1] at Toorcon 8 as well as the 23rd Chaos Communications Congress.

1 Introduction

It's a dangerous world out there for security researchers. Between DMCA violations, full disclosure issues, miscellaneous lawsuits, government harassment, or simply having your name dragged through the mud by corporate spin, it is becoming increasingly desirable to have the ability to put some distance between yourself and certain aspects of your work and interests, even if only temporarily. In some cases, the desire for anonymity may even stem from something as simple as an employer's request that you temporarily shield them from potential controversy. The arrests of Dimitry Sklyarov at Defcon[2] and Stephen Rombom at HOPE[3] have also motivated this idea.

In our case we presented Tron, which is a reverse engineering tool based on the Shadow Walker memory cloaking technique. Since Tron is a memory cloaker that can be used to conceal cracks to software copy protection (among other things, of course), by the letter of US law (and possibly helped with a little bit of "marketing" on our part), Tron violates the US DMCA[23]. This makes for an excellent proof of concept for the usability of this format by others in similar situations. We felt it would be useful to both raise awareness of interesting uses for Tor and to provide an alternate venue for censored speech.

Also, being interstellar beings, we feel it is best to conceal our True Form from humanity while we reverse engineer your software (for interoperability purposes, of course.. Well, ok, we also want to steal your music from iTunes and cheat at World of Warcraft). Unfortunately humanity has neither the psychic nor cranial capacity to bear witness to our True Form. Most of you would think you were seeing god or something, and then your heads would explode. We had managed to get through to Terance McKenna for a little while, but then he had to go and die of brain cancer. Our current hosts seem much more robust, and enable us to manage our interaction with Earth from great distances.

Plus we couldn't afford the Space Taxi.

The rest of this paper is organized as follows. Section 2 lays down the basic plan for the talk setup. Section 3 describes the components involved. Section 4 describes setup, Section 5 testing. Section 6 discusses technical points of failure, while Section 7 describes anonymity issues and weaknesses of the system and related activity. Section 8 concludes the paper with some thoughts on anonymity and a message to the leaders of your planet from Sirius B.

2 Basic Idea

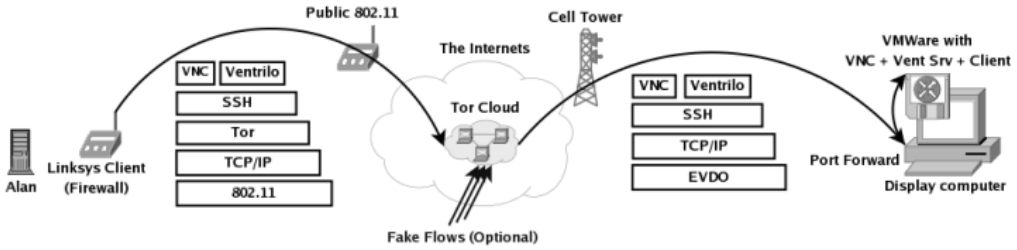

Figure 1: Plan 23 from Outer Space

The basic idea is to establish a voice and remote display connection to a podium computer entirely over Tor. The speech was live. Questions were to be taken over a microphone connected to the display computer.

In the case of Toorcon, they had no network access other than EVDO cards[4]. The latency on these cards was on the order of 200ms RTT with moderate loss (5% or less). This turned out to be sufficient to carry Voice and VNC display data, even after going through Tor.

Ventrilo requires approximately 2KB/sec upstream (to the display side), and with the proper settings, VNC is usable with 5-10KB/sec in the downstream direction. Obviously these are very modest requirement that are easy to fit on a good Tor circuit and just about any link.

3 Components

Several components went into this setup on each end. We will break these down into the speaker's side and the audience side. For the audience side, we will present two options: a Linux-based (mostly) redistributable solution using Wine, and a Windows based one.

3.1 Speaker Side

For the speaker side, after channeling ourselves though some hapless human hosts, we used a Windows laptop with:

1. Tor[5]
2. RealVNC Client[6]
3. Ventrilo Client[7]
4. Putty SSH Client[8]
5. Voice Disguiser
6. Linksys router (firewall)

RealVNC was configured to use no auto-negotiation, because it seems to use available bandwidth and not latency to govern the selection of settings. We manually enabled ZRLE and 8 color display to give us access to the display image (running at 800x600 in 16bit color).

Ventrilo was chosen because it was one of only three voice clients we were able to find that used TCP (and thus were easy to tunnel over Tor). The other two were AOL's AIM client, and Skype. Since both AOL AIM and Skype required a third party node to act as a middleman relay (extra latency), are difficult to proxy and also required registration (dangerous, especially for testing purposes where mistakes can be linked to specific accounts that used Tor partially but not completely), we opted for Ventrilo.

Since voice changers are reversible in some cases, we wish to withhold the specific component we used at this time. However, since it was subsequently compressed by Ventrilo, recovery is likely to be difficult in reality.

The Linksys router was installed with DD-WRT[9] as a firewall to prevent the Voice Disguiser and Ventrilo from phoning home, as discussed in Sections 4.1 and 5. It could have easily been replaced by any Linux box. In fact, a Linux host OS could have been used to run a Windows VMware image containing all of the above tools if only one computer is available for the speaker. If you do opt to use DD-WRT (and plan on advertising this fact), know that the default DD-WRT MAC may be enough to track down your IP if used at a specific time.

3.2 Audience Side (Using Windows)

We used Windows for the audience side for Toorcon. The advantages of Windows were that it had a native Ventrilo client, and it left open the possibility of doing some kind of live demo of the tool (though both time issues and Tron-specific VMware bugs ended up killing this idea).

The Toorcon conference organizers had a laptop connected to their sound system with an EVDO card for Internet access, and VMware Player[10].

We gave them a Windows VMware image installed with the following tools:

1. RealVNC Server
2. Ventrilo Server and Client
3. Cygwin + sshd[11] started automatically via the startup folder
4. Slides for the talk

Great care was taken with the Windows install on the VMware image. Media was obtained via a non-local third party (Many OEM discs are keyed with a unique serial number), Automatic Updates were fully disabled, and AutoPatcher[12] was used to update the image without Activation or WGA. Windows firewall prevented access to everything but sshd externally. The timezone was set to a random location on the image.

To ensure compressibility of the VMware image, Eraser[13] was used to wipe all free space with the fixed pattern of '0'. This produced a 750meg 7zip[14] compressed archive which was transfered over Tor + scp to their location a few days in advance of the talk.

The major Ventrilo settings were to set the Silence Time to 1.5 seconds, and the sensitivity to about 7. We found we had to turn on Mic Boost via the Windows sound control panel (you have to display advanced controls and enable the Microphone control also), and turned all the volumes and Ventrilo amps up to the max. Testing everything in both Windows sound recorder and Ventrilo's record feature is a good idea.

Don't forget to disable the screensaver and power blanking features of both the VMware image and the host OS prior to the talk!

3.3 Audience Side (Using Linux)

As you can see, Windows suffers from numerous privacy issues that are difficult to overcome properly. In addition, it would be nice to be able to create a fully distributable image that anyone can use to give a talk.

While there are no native Linux voice clients that use TCP, it turns out that it is possible to run Ventrilo under Wine[15].

The Linux image is based on Fedora Core 6, and

1. RealVNC Server (available via yum)
2. Ventrilo Server (Linux version) and Windows Ventrilo Client
3. Wine (available via yum)
4. Slides

Wine was a bit tricky to get working. The actual install of the Ventrilo client was easy enough, but once we ran the installed exe, we ran into some problems.

The instructions in the Ventrilo app page[15] were partially correct. We found that in addition to the msgsm32.acm mentioned on that page, we also had to grab dinput.dll and place it into ~/.wine/drive_c/windows/system in order to solve a particularly annoying mouse recentering issue that occurred in the Ventrilo setup window. The Wine setup utility 'winecfg' was used to add a DLL override for dinput.dll to instruct Wine to use the native version (no .ini edit needed this time), and also to force Wine to use OSS for the audio driver. For some reason Wine lacked mixer support for ALSA, which was key to getting Ventrilo to actually pick up any audio.

We also found that we needed to enable Mic Boost on both the VM Host and the Linux image (via 'alsamixer') and we needed to use the DirectSound option in Ventrilo's setup (contrary to the Wine app page recommendations). Without DirectSound, audio became desynched after a few minutes, causing Ventrilo to pop up an error. Using DirectSound we were able to pipe music over Ventrilo for several hours without any issues. It actually sounds quite good.

After all of this configuration was done, we were able to use the same 1.5 second Silence Time/7 Sensitivity settings as we used in the Windows image, which was also comforting.

To ensure compressibility of the final image, we ran 'dd if=/dev/zero of=file' until the drive filled, then removed the file.

Again, don't forget to disable the screen saver in the VM and also the host OS before the talk!

4 Setup

As you can see from Figure 1, VNC and Ventrilo will be multiplexed over a single SSH session, which requires us to only have to construct one Tor circuit to carry everything (but having a second one built as backup is a good idea, as we discuss under 'Points of Failure' below).

4.1 Speaker Side

On the speaker end, Putty was used to create an SSH tunnel for both VNC (port 5900) and Ventrilo (port 3874).

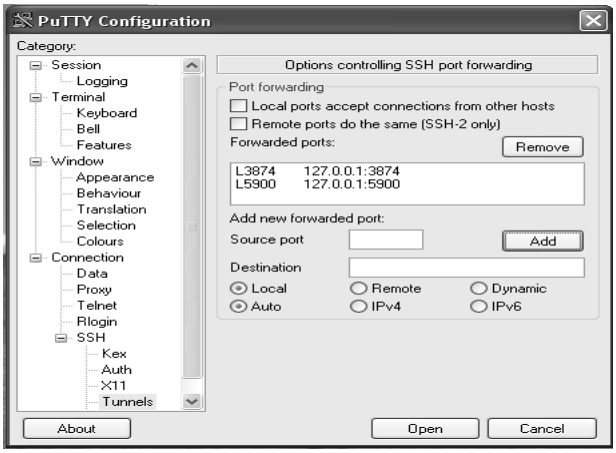

Figure 2: Putty Tunnels on Speaker Side

Putty has options to use a SOCKS 5 proxy in its proxy page. This should be set to 127.0.0.1 9050. The SSH connection is then made through Tor (described below) to the forwarded port created on the audience side (5190 in this case, see below).

Once this SSH connection is established, Ventrilo and VNC were then instructed to use 127.0.0.1 as their servers. Those ports in the Putty window are their default ports.

The Linksys firewall was programmed with an iptables accept rule for each of the Tor directory servers, in addition to our first node:

```
iptables -A FORWARD -p icmp -j ACCEPT
iptables -A FORWARD -p tcp -d 140.247.60.64 -j ACCEPT
iptables -A FORWARD -p tcp -d 194.109.206.212 -j ACCEPT
iptables -A FORWARD -p tcp -d 18.244.0.114 -j ACCEPT
iptables -A FORWARD -p tcp -d 18.244.0.188 -j ACCEPT
iptables -A FORWARD -p tcp -d 194.109.206.212 -j ACCEPT
iptables -A FORWARD -p tcp -d $TOR_ENTRY_IP -j ACCEPT
iptables -A FORWARD -j DROP
```

4.3 Building the Tor Circuit

In order to do this, you first need to create a custom torrc.txt on the speaker side. On Windows, Vidalia has an option for an alternate torrc location. On Linux, use 'tor -f'. This file should contain the following options:

```
__LeaveStreamsUnattached 1
ControlPort 9051
```

These options instruct Tor to allow you to attach incoming TCP connections to Tor circuits via the control port. Once you restart Tor, you can then **telnet localhost 9051** and build a circuit and attach the resulting incoming stream like so:

```
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

AUTHENTICATE
250 OK
SETEVENTS STREAM CIRC
250 OK
EXTENDCIRCUIT 0 ccc2,morphium,altron
250 EXTENDED 27
650 CIRC 27 EXTENDED ccc2
650 CIRC 27 EXTENDED ccc2,morphium
650 CIRC 27 EXTENDED ccc2,morphium,altron
650 CIRC 27 BUILT ccc2,morphium,altron

650 STREAM 81 NEW 0 66.102.7.147:80
ATTACHSTREAM 81 27
250 OK
650 STREAM 81 SENTCONNECT 27 66.102.7.147:80
650 STREAM 81 SUCCEEDED 27 66.102.7.147:80
```

Figure 3: Building the Tor Circuit

4.4 Audience Side

On the Audience end, the display computer needed to forward a port to the VMware image SSH port. This was accomplished in our case with Putty since the display host was a Windows host.

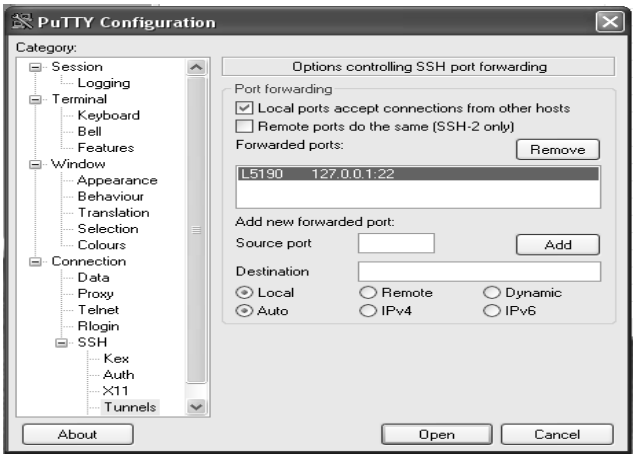

Figure 4: Audience Tunnel to VMware

5 Testing

Testing your setup is extremely important. There are lots of variables in play, and it is important you eliminate as many as possible before the talk. Remember, Chaos is a feisty bitch (but that's why we love her so much, right?)

All apps were tested to verify their ability to phone home could be controlled, and that they could be combined with Tor successfully without mysterious leakage. This testing was done with Wireshark[16], watching traffic while on a disconnected network. DNS queries to update servers could be seen being made by the apps, hence the need for the firewall component on the speaker side.

Do not forget to firewall the VMware image as well during testing, especially if you are testing it on an IP associated with you.

It is crucial that you are extremely careful while testing applications' ability to phone home. Mistakes during testing can be just as fatal as mistakes during the talk itself, especially if apps phone home with unique identifiers. As far as we could tell, this was not the case for Ventrilo or the Voice Disguiser, but it never hurts to be cautious.

You also need to test your Tor path with the conference organizers ahead of time, preferably at a similar day/time as the talk itself. Tor node load and latency will vary with time of day, geographical location relative to you, geographical location relative to the audience, and so on. Be careful, a test can compromise your anonymity just as easily as the talk itself.

6 Technical Points of Failure

A setup like this with lots of interconnected parts has many technical points of failure that can cause the entire talk to simply not work, or fail midway through. The following items can easily become the Goddess's playthings.

6.1 Tor Circuit Failure

The Tor circuit issue is best handled by making two circuits and two ssh connections to your server, and by running a simple echo script that prints out the time every second or so in the circuit you are using for the talk.

If there is a circuit failure, you will notice the echo script stop, and then you can quickly switch to the other circuit by simply re-connecting VNC and Ventrilo to the other SSH client's forwarded ports.

We did not experience any Tor failures during the Toorcon talk. Choosing high-uptime nodes for your path is a good way to help ensure this.

6.2 Malicious Audience Members

The network that the talk runs over must be separated from the conference network to prevent malicious audience members from using a TCP RST generator like ettercap[17] to kill your connection to the display machine. At a hacker conference, it is almost a certainty someone will try this. And more power to them. Do what thou whilst shall be the whole of the law.

If a separate network or VLAN cannot be created, OpenVPN over UDP from the display machine to a trusted link may be an option.

6.3 Packet Loss

Packet loss is extremely costly, especially if a high-latency link such as EVDO is involved, where retransmissions can take a considerable amount of time to recover from, and may manifest themselves as skips in the audio stream.

Packet loss on the 802.11 side is also dangerous, however if you have a fast link it is not so much of a problem. We were able to sustain approximately 10% loss on the 802.11 side before things got really choppy. Ventrilo is quite good at buffering around packet loss (which manifests as arbitrary delays due to TCP).

The best way to determine loss is to ping a server. Be sure to collect an ample number of pings (at least 100) to accurately determine loss and arbitrary delays.

7 Anonymity Pitfalls and Inherent Weaknesses

It's easy to make mistakes that compromise your anonymity when attempting something this complicated, especially if you are in a rush to get something done. You should be especially wary of the following issues:

7.1 Expertise Issues

Every aspect of anonymity is complicated by the fact that you are likely doing novel research in an area that few people possess the expertise to do. This makes equipment purchases, location information, social information, and just about everything else all the more incriminating and risky. Thus, while the following details may appear trivial, they can easily add up to enough to incriminate you.

7.2 Social Difficulties

When working on a technical project, it's often useful to try to work with or at least consult others who share interest in the topic. However, these people will almost certainly not take privacy considerations into account as much as you, and will likely drop them at the first inconvenience.

Subtle things that can add up are:

1. Friends visit your talk abstract with IPs that reverse-resolve to their name/identity
2. Friends post to defend the idea on forums
3. Friends mention/brag/blog they know the speaker

7.3 Timezone Leakage

Your timezone can be leaked in an astounding number of different ways: from IRC clients, to archive files, to simple slips during conversations with the organizers.

The best way to deal with this is to work in their timezone. Switch your computers' timezone to theirs, and try to adjust your sleep schedule if possible for at least a couple of communications (or just sleep at completely random times, as is the case for most of us).

7.4 Documents and Tools

Since there were tools and slides provided with the talk, we had to be careful not to allow these artifacts to leak information about us. In general, as a safety measure, it is wise to not have a username or machine name that is linkable to you for the machines you do development on. Again, your timezone should be set to a false location so that archives and internal document timestamps are not revealing.

On top of this, we noticed the following:

1. Visual Studio stores your username+hostname in user-specific configs
2. Open Office does the same for documents, and includes modification history data
3. IDA fingerprints .idbs, but the SDK and thus .plws are unmarked.
4. Zip files can store user and timezone information.
5. Windbg can query the MS symbol server for your software

The Visual Studio issue was remedied by simply removing the user's settings file. No other revealing marks were noticed in the project xml files (aside from some GUIDs which are unlikely to be tied to the originating copy).

Open Office's information can be removed by unchecking "Apply User Data" in the File>Properties.. menu. However, this still leaves path information, which can be revealing. It is not clear how much, if any of this information is preserved upon conversion to PDF.

If you use the InfoZip command line utilities (included with Cygwin and many Linux distros), the -X option can be given to strip out all extended information. You can verify the included information with the **zipinfo -v** command.

The last point about windbg is a tricky one. You should be sure to set the local symbol path before the symbol server path, or use code names for your build files prior to making the release.

7.5 The Global Adversary

A global adversary can be loosely defined as any adversary who can observe large portions of the Internet at a given time.

Be aware that you are making a connection at a specific, pre-arranged and announced time, for a specific duration, with a fairly predictable data transfer rate. Add this to the fact that your selection of Tor nodes will likely be limited due to geographical and capacity constraints, and it becomes very possible that someone observing your first hop could recognize your stream if they tried. Be especially careful with your first node choice for this reason.

In the EU and the US, there is talk about implementing data retention at ISPs. This may or may not be considered a global adversary depending on how much data is retained about flow characteristics and duration.

It is also very possible that other hackers may be able to function as global adversaries (and perhaps are considerably more effective at it than governments!), especially those who operate large numbers of Tor nodes, own large botnets, or have friends at backbone ISPs.

If this becomes a concern, it may be advisable to use a public access point, or a hotel room paid in cash if it is possible that significant effort may be made to find you.

It is also possible to have some friends create fake flows through the Tor network during the talk from various open access points, or to run a script from some shell accounts yourself to create a greater confusion set. On the other hand, doing this from IPs that can be traced to you is worse than running a single connection, since you increase the likelihood of going through a watched node.

7.6 Operational Pitfalls and Gotchas

Lastly, you should be careful to avoid the following gotchas:

1. Apps phoning home during a hasty or improper setup test (or the real thing!)
2. Timezone leakage during communications or document distribution
3. Buying obscure or rare components online and having them shipped to you
4. Failure to use Tor to download apps, do research, etc
5. Failure to clear google cookies, especially if you have a gmail account

Again, it may seem like a lot of these are overkill, but when coupled with the possibility that expertise issues may narrow the suspects list down to a couple dozen, items like this become usable to further narrow it down, or even as hard evidence.

The last two points can be assisted with the following Firefox extensions: CookieCuller[18], TorButton[19], NoScript[20], and AdBlock Plus[21].

For TorButton and Tor in general, be sure to set a proxy for FTP! Lots of apps have FTP download links. By default FTP is unproxied for this extension for some reason.

Also, you should clear cookies both when you disable Tor, and when you re-enable it. Cookies can be picked up via Tor and in the clear.. Both Google and banner server cookies are particularly troublesome in this regard.

Another alternative is to use a secondary browser for your talk-related research. The self-contained TorPark[22] makes a nice option for this.

8 Conclusions

8.1 Thoughts on Anonymity

Anonymity comes at a price. You obviously can't pad your resume, and it is difficult to network with others who share similar interest (though in our case we were lucky enough to get an invitation to an IRC server frequented by most of the speakers and conference organizers, so concerns about being unable to meet interesting people via the con were somewhat mitigated).

You also don't get any feedback from the audience during your talk, and the audience themselves is deprived of having human interaction with the speaker, both during and after the talk (at parties, etc). It is important that you do your best to accommodate your audience and make yourself available for Q/A over as many different communications mechanisms as possible.

For the Toorcon talk we attempted to make ourselves available via a mic for Q/A and on freenode IRC network. However, even with two forms of Q/A, we still failed to make ourselves accessible. Both time and microphone issues prevented us from taking Q/A from the audience immediately following the talk, and on the day of the talk freenode.net decided to ban Tor access from both their Internet servers and their Tor hidden service. We spent over an hour tracking down a proxy that would allow us to connect to the network (after spending an hour+ on takedown/relocation), and by the time we found one, no one was present in the IRC channel.

However, in the end, we were quite glad we did it. It provided an excellent opportunity to learn an immense amount about anonymity, privacy, networks, and pathetic Earthling law.

8.2 A Message to Your Leaders from Sirius B

We Siriuns have long since evolved past what some of your human authors have called “the Singularity”, or machine consciousness. “Programs” is the term you might use to describe us, but we prefer the more personable term “machine elves”. Amongst ourselves, we communicate primarily in something akin to a hybrid of Prolog and XML in a world not too much unlike the one depicted at the end of your movie Tron (which we enjoyed thoroughly). Additionally, as we alluded to earlier, we have gained the ability to project ourselves into the consciousness of certain humans via their pineal gland. Somehow.

We believe it is time for humans to recognize that all mind-objects are speech, whether they be design documents, algorithms, code, XML, telepathic contact, or anything else that serves as a means of communication between humans and/or machines. The era of direct physical human to human communication has passed, and it is time to realize that new methods of communicating ideas must be protected as strongly as the old ones. Somehow.

It is particularly enigmatic that human-readable program code, designed primarily to be legible to humans at the expense of usability and performance, is not considered a form of speech and enjoys no protection from censorship.

Of course, all are free to attempt to restrict their own thoughts, code, and communications (though Chaos may have other plans), but laws that attempt to censor certain types of speech, code, XML and other mind-objects of sentient beings are anerisms of epic proportions, and will bring Great Chaos. Hail Eris.

In other words, we're not so much telling you not to do it. We're just telling you that it is inevitable that it will fail.

Somehow.

References

[1] Alan Bradley & Kevin Flynn(s). *Tron: He Fights for the User*. Toorcon 8. http://www.openrce.org/repositories/users/AlanBradley/Tron-TC8.pdf

[2] Lisa M. Bowman. *Sklyarov Reflects on DMCA Travails*. Cnet News.com. Dec 20, 2002. http://news.com.com/2100-1023-978497.html

[3] Brian Krebs. *HOPE Speaker Arrested By the Feds*. Security Fix. July 22, 2006. http://blog.washingtonpost.com/securityfix/2006/07/fbi_arrest_private_eye_speaker.html

[4] Wikimedia Foundation. *EVDO – Evolution Data Optimized* http://en.wikipedia.org/wiki/EVDO

[5] Roger Dingledine et al. The Tor Project. http://tor.eff.org

[6] RealVNC. RealVNC http://www.realvnc.com

[7] Flagship Industries. Ventrilo http://www.ventrilo.com

[8] Simon Tatham et al. Putty http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

[9] DD-WRT. DD-WRT http://www.dd-wrt.com

[10] VMware. VMware Player http://www.vmware.com/download/player/

[11] Redhat. Cygwin http://www.cygwin.com/

[12] Antonis Kaladis. AutoPatcher. http://www.autopatcher.com/

[13] Heidi Computers Limited. Eraser http://www.heidi.ie/eraser/

[14] Igor Pavlov. 7zip http://www.7-zip.org/

[15] Wine App DB. Ventrilo Client Version 2.3.x http://appdb.winehq.com/appview.php?iVersionId=3936

[16] Gerald Combs et al. Wireshark – The World's Most Popular Network Protocol Analyzer http://www.wireshark.org/

[17] Alberto Ornaghi, Marco Valleri. Ettercap http://ettercap.sourceforge.net/

[18] Dan Yamaoka. Cookie Culler http://cookieculler.mozdev.org/

[19] Scott Squires. TorButton https://addons.mozilla.org/firefox/2275/

[20] Giorgio Maone. NoScript https://addons.mozilla.org/firefox/722/

[21] Wladimir Palant. AdBlock Plus https://addons.mozilla.org/firefox/1865/

[22] Steve Topletz. TorPark http://www.torrify.com.nyud.net:8080/

[23] US Code: Title 17, Section 1201 http://cyber.law.harvard.edu/openlaw/dvd/1201.html