# InfraGard Technology Tool Kit Disclaimer Notice:

# Created by Deborah Watson, Technology Tool Kit Program Development Manager, InfraGard - Houston Chapter; adapted from content created by Dave Armstrong, dave@bindshell.net

# IN NO EVENT SHALL INFRAGARD BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND/OR DOCUMENTATION, EVEN IF INFRAGARD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# INFRAGARD SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE AND/OR ACCOMPANYING DOCUMENTATION, IF ANY, PROVIDED HEREUNDER IS PROVIDED "AS IS". INFRAGARD HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

# Distributed Networked John

This is a client/server framework that wraps around a slightly modified version of John The Ripper.   
In contrast to the MPI version of John, dnetj allows the use of nodes which are of differing speeds and for nodes which do not run 24/7.  
This tool was written for a number of reasons, firstly the MPI version requires an MPI installation on each node, and for the nodes to be configured together and be roughly the same speed. Also, although other distributed password crackers exist (such as djohn or medussa) they all have their own limitations.  
The server loads a set of password hashes, and splits the available keyspace into "work units" of a configurable size. The clients connect and retrieve the hashes, as well as a set of work units to process. Once a client has processed some work units, it connects back to the server to submit the completed units as well as any passwords which have been cracked.  
Possible uses include eg, running as a background task on all the workstations in an office.  
Note, this tool is at an early stage of development and is likely to be very buggy, although it is functional. Bug reports and/or patches are strongly encouraged.

**Developer:** [John Anderson](mailto:john@ev6.net)

**Mailing List**

There was a mailing list for Dnetj available, you can subscribe to it by writing a mail containing the word "subscribe" in the body of the mail, and sending it to john-mpi-subscribe@bindshell.net. As this list is currently fairly low traffic, it is shared with the john-mpi list. It is likely that is has been retired as bindshell.net is retired and due to be taken offline.

**Support**

For support queries relating to Dnetj, expect to be on your own. Do not contact the original authors of John The Ripper or use the public mailing lists on openwall.com for issues which are specific to the modified versions.

**Supported Platforms**

Dnetj has been tested on the following systems:

* Linux/x86
* Linux/amd64
* Linux/sparc
* MacOSX/PPC
* MacOSX/Intel
* Solaris/x86

**Features**

* Distributed client/server model, any number of clients can be supported and can be brought up and down at will
* Support for any cipher supported by John 1.7.2 (additional cipher patches should be able to be applied normally)
* Keep the changes to John to a minimum, so that patches/updates can still be applied without too much fuss
* Uses the same optimized encryption routines as John
* Cracked hashes are stored in the standard john.pot format, so they can be displayed with john -show
* Password files sent to clients are sanitized (only the hash is sent, other fields from the password file are removed)
* Capability for auto client registration
* Code is intentionally kept clean to aid porting

**Changes**

0.2.5 - Initial public release

**Known Issues**

* Clients will sometimes crash if unable to connect to the server for a long period of time.
* Work unit size is limited to a 32bit integer number of crypts (ie: 4294967296)
* Node performance calculations wrap once the node has performed more than 4294967296 crypts, so nodes may appear to be much slower than they truly are.
* Doesn't work with NTLM, as the NTLM hash is stored in a different field of the password file.
* Traffic is sent in plain text (this makes debugging easier at this early stage of development)
* Makefile is very basic, and has no configure script, compilation on Solaris requires adding -lnsl -lsocket to the compile command.

**Download**

A safe version of Dnetj can be obtained from the InfraGard site at[need to insert link] [dnetj-0.2.5.tar.gz](http://bindshell.net/tools/dnetj/dnetj-0.2.5.tar.gz) (md5sum: 88a0270ef8579e977c77c1d73df5b702)

**Documentation**

There is currently a rudimentary "INSTALL" file located in the distribution archive.

Currently the client/server must be compiled separately from the bundled version of John.

The sourcecode to John is located in the src directory, and this can be compiled in the same way John is normally compiled. One caveat is that for a Dnetj server installation, the Makefile needs to be edited and -DMASTER added to the CFLAGS line.

The client/server source is located in the "dist" subdirectory, and can be compiled with a simple "make". This will create 2 binaries, "client" and "server".

The client or server binary should be moved to the "run" subdirectory, alongside the john binary. Also copy the file conf or conf-client into the run directory and edit it accordingly, most of the options within the configuration file are self explanatory or explained by comments.

To execute, start either the client or server with it's corresponding configuration file as the first argument, eg:  
./server conf  
./client conf-client