Hackersh Documentation

Release 0.3.dev0

Itzik Kotler

October 12, 2013

CONTENTS

Welcome to Hackersh's documentation. This documentation is divided into different parts. I recommend that you get started with *Installation* and then head over to the *Quickstart*. If you'd rather dive into the internals of Hackersh, check out the *API Reference* documentation.

Hackersh uses Pythonect as its scripting language. To learn more about Pythonect, visit Pythonect's Web site <http://www.pythonect.org>

Note: This is the main documentation for the Hackersh project. The contents of this site are automatically generated via Sphinx based on the Python docstrings throughout the code and the reStructuredText documents in the doc/ directory of the git repository. If you find an error in the documentation, please report it in the bug tracker here, or even better, submit a pull request!

CHAPTER

ONE

USER'S GUIDE

This part of the documentation, which is mostly prose, begins with some background information about Hackersh, then focuses on step-by-step instructions for getting the most out of Hackersh.

1.1 Introduction

Read this before you get started with Hackersh. This hopefully answers some questions about the purpose and goals of the project, and when you should or should not be using it.

1.1.1 Philosophy

"The whole is greater than the sum of its parts." Aristotle

1.1.2 What is Hackersh?

Hackersh ("Hacker Shell") is a free and open source command-line shell and scripting language designed especially for security testing.

Just like Linux system administrators are using shell scripting to automate tasks:

```
> /sbin/ifconfig | /bin/grep "inet addr:" | /usr/bin/cut -d: -f2 | /usr/bin/awk '{ print $1 }'
127.0.0.1
```

Hackersh aims to help security testers to automate their tasks:

```
> /sbin/ifconfig | /bin/grep "inet addr:" | /usr/bin/cut -d: -f2 | /usr/bin/awk '{ print $1 }' | ipv4

Properties:
-----------

+--------------+-----------+
| Property     | Value     |
+--------------+-----------+
| Ipv4_Address | 127.0.0.1 |
+--------------+-----------+
| Name         | 127.0.0.1 |
+--------------+-----------+
| Service      | HTTP      |
+--------------+-----------+
| Proto        | TCP       |
```

```
+--------------+-----------+
| Port         | 80        |
+--------------+-----------+

Graph:
------

127.0.0.1 <via str>
`-127.0.0.1 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
    `-Found #4 Vulnerabilities <via nikto>

Vulnerabilities:
----------------

+-------------------------------------------------------------------------------+-----------------
| VULNERABILITY DESCRIPTION                                                     | URL
+-------------------------------------------------------------------------------+-----------------
| ETag header found on server, inode: 436622, size: 177, mtime: 0x4e22ce6d50080 | http://localhost
+-------------------------------------------------------------------------------+-----------------
| Allowed HTTP Methods: GET, HEAD, POST, OPTIONS                                | http://localhost
+-------------------------------------------------------------------------------+-----------------
| /server-status: This reveals Apache information. Comment out appropriate line in | http://localhost
| httpd.conf or restrict access to allowed hosts.                               |
+-------------------------------------------------------------------------------+-----------------
| /icons/README: Apache default file found.                                     | http://localhost
+-------------------------------------------------------------------------------+-----------------
```

It is written in Python and uses Pythonect as its scripting engine.

Continue to *Installation* or *Quickstart*

1.2 Installation

This part of the documentation covers the installation of Hackersh. The first step to using any software package is getting it properly installed.

1.2.1 Installing Hackersh

Hackersh requires Python version 2.6 and greater, but it will not work (yet) with Python 3. Dependencies are listed in `setup.py` and will be installed automatically as part of any of the techniques listed below.

Note: Hackersh will ***NOT*** install 3rd party security tools as part of its installation. You have to manually download and install each and every security tool that you wish to use in Hackersh. Alternatively, you can install Hackersh in a Linux distribution such as BackTrack, Kali, or Pentoo and enjoy the already installed tools.

Distribute & Pip

Installing Hackersh is simple with pip:

```
$ pip install hackersh
```

or, with easy_install:

```
$ easy_install hackersh
```

Note: Using easy_install is discouraged. Why? Read here.

1.2.2 Download the Source

You can also install Hackersh from source. The latest release (0.3) is available from GitHub.

- tarball
- zipball

Once you have a copy of the source, unzip or untar the source package, cd to the new directory, and:

```
$ python setup.py install
```

To download the full source history from Git, see *Source Control*.

Staying Updated

The latest version of Hackersh will always be available here:

- PyPi: http://pypi.python.org/pypi/hackersh/
- GitHub: http://github.com/ikotler/hackersh/

When a new version is available, upgrading is simple.

```
$ pip install hackersh --upgrade
```

1.3 Quickstart

Eager to get started? This page gives a good introduction to Hackersh. It assumes that:

- You already have Hackersh installed. If you do not, head over to the *Installation* section.
- You are familiar with Pythonect. If you aren't, head over to the Pythonect Tutorial

1.3.1 Using the Shell

Once Hackersh is *installed*, you can run it from the command-line like this:

```
$ hackersh
Hackersh Version 0.3.0
Copyright (C) 2013 Itzik Kotler
Visit http://www.hackersh.org for updates.

********************************************************************************
*                                                                              *
*                                                                              *
*  Welcome to Hacker Shell Version 0.3.0                                       *
*                                                                              *
*                                                                              *
********************************************************************************
```

```
>
```

Typing `help` or `?` once in the Hackersh shell prompt will list the commands available to you.

```
> help

Shell Builtin Commands
======================

    Command        Description
    -------        -----------
    ?              Display information about builtin commands
    exit           Exit the shell
    help           Display information about builtin commands
    info           Queries the supplied component or components for information
    quit           Exit the shell
    show           Displays components
```

Type `help X` to find out more about the command `X`.

```
> help show
Usage: show [all|internal|external|root]
```

Or pass the command-line option `-h` or `--help` to the command:

```
> show -h
Usage: show [all|internal|external|root]
```

When running interactively (i.e. when commands are read from a tty), Hackersh will employs the GNU readline library to provide some useful command line editing facilities, as well as to save command history. Pressing `Tab` will autocomplete commands, components, directories, files, and more.

```
> info ipv4_
ipv4_address  ipv4_range
```

Pressing `Up` and `Down` will navigate through all the history commands entered at the prompt.

The command history is saved in the `.hackersh_history` file in your home directory between different invocations of the shell.

1.3.2 Running Commands

Hackersh runs commands like other shells:

```
> /bin/ping -c 3 192.168.1.110

PING 192.168.1.110 (192.168.1.110) 56(84) bytes of data.
64 bytes from 192.168.1.110: icmp_req=1 ttl=64 time=0.224 ms
64 bytes from 192.168.1.110: icmp_req=2 ttl=64 time=0.064 ms
64 bytes from 192.168.1.110: icmp_req=3 ttl=64 time=0.449 ms

--- 192.168.1.110 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.064/0.245/0.449/0.158 ms

> /usr/bin/nmap 192.168.1.110 -p 80

Starting Nmap 6.25 ( http://nmap.org ) at 2013-09-22 17:08 IDT
```

```
Nmap scan report for 192.168.1.110
Host is up (0.00026s latency).
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:AD:A8:E7 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds

> /usr/bin/nikto -host 192.168.1.110 -port 80

- Nikto v2.1.4
---------------------------------------------------------------------------
+ Target IP:          192.168.1.110
+ Target Hostname:    192.168.1.110
+ Target Port:        80
+ Start Time:         2013-09-23 17:09:46
---------------------------------------------------------------------------
+ Server: Apache/2.2.16 (Debian)
+ Retrieved x-powered-by header: PHP/5.3.3-7+squeeze15
+ Apache/2.2.16 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42 (final rele
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/
+ OSVDB-12184: /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive
+ OSVDB-3268: /files/: Directory indexing found.
+ OSVDB-3092: /files/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6448 items checked: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2013-09-23 17:09:55 (9 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

If a command starts with / (slash), ./ (dot slash), or ../ (dot dot slash) it is executed as a system command.

1.3.3 Running Components

Hackersh comes with a library of components for security testing. The components are like building blocks. They offer various assemblies with their parts being interchangeable.

Hackersh runs (and pipes) components like commands: you type a component name, followed by its arguments.

```
> "192.168.1.110" | ipv4_address | nmap -p 80 | nikto

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Proto        | TCP           |
```

```
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap -p 80>
    `-Found #10 Vulnerabilities <via nikto>

Vulnerabilities:
----------------

+-----------------------------------------------------------------------------------------------+---------
| VULNERABILITY DESCRIPTION                                                                     | URL
+-----------------------------------------------------------------------------------------------+---------
| Retrieved x-powered-by header: PHP/5.3.3-7+squeeze15                                          | http://1
+-----------------------------------------------------------------------------------------------+---------
| Apache/2.2.16 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42      | http://1
| (final release) and 2.0.64 are also current.                                                  |
+-----------------------------------------------------------------------------------------------+---------
| DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-      | http://1
| us/library/e8z01xdh%28VS.80%29.aspx for details.                                              |
+-----------------------------------------------------------------------------------------------+---------
| /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive        | http://1
| information via certain HTTP requests that contain specific QUERY strings.                   |
+-----------------------------------------------------------------------------------------------+---------
| /files/: Directory indexing found.                                                            | http://1
+-----------------------------------------------------------------------------------------------+---------
| /files/: This might be interesting...                                                         | http://1
+-----------------------------------------------------------------------------------------------+---------
| /img/: Directory indexing found.                                                              | http://1
+-----------------------------------------------------------------------------------------------+---------
| /img/: This might be interesting...                                                           | http://1
+-----------------------------------------------------------------------------------------------+---------
| /icons/: Directory indexing found.                                                            | http://1
+-----------------------------------------------------------------------------------------------+---------
| /icons/README: Apache default file found.                                                     | http://1
+-----------------------------------------------------------------------------------------------+---------
```

You can mix between Hackersh components and any third-party binaries (e.g. cat, grep, and etc.) as long as the binaries output (via stdout) something meaningful:

```
> /bin/cat /etc/hosts | /bin/grep "127.0.0.1" | /usr/bin/awk '{ print $1 }' | /usr/bin/tr -d '\n' | :

127.0.0.1 <via str>
`-127.0.0.1 <via ipv4_address>
  +-80/tcp (HTTP) <via nmap_result_#1>
  `-22/tcp (SSH) <via nmap_result_#0>
```

1.3.4 Getting Help

To get help on a specific component, use the built-in `info` command:

```
> info ipv4_address
Component: ipv4_address
Version: 0.1.0
Source: /usr/local/lib/python2.7/dist-packages/Hackersh-0.3.dev0-py2.7.egg/hackersh/components/interr
Type: RootComponent
Provided by: Itzik Kotler <xorninja@gmail.com>
Filter: None
Query: None
Description:
   Convert String to IPv4 Address
```

To get a specific component usage, pass the command-line `-h` or `--help` to it:

```
> nmap -h
...
...
...
```

Don't worry if the application don't take `-h` or `--help` options. Hackersh will automatically map it to the correct command line option.

1.3.5 Debugging

Hackersh offers two debugging options: Shell Debugging and Component Debugging. These options can be enabled together, or enabled separately.

To debug the shell simply pass the command line option `-v` to increment the verbosity:

```
$ hackersh -vvv
```

To debug a specific Hackersh component, just pass: `debug=True` to it and see what the input and output strings are:

```
> "127.0.0.1" | ipv4_address | nmap('-p 80', debug=True) | nikto
```

1.3.6 Return Values

Most Hackersh components take and output Context. Context is a directed graph where each node is a dictionary. The node dictionary contains key/value pairs that contain the properties of a single component execution result. Each node points to its *successor*, which is another properties dictionary of another single component result that is waiting on it to complete.

Hackersh starts with an empty Context and after a successful execution it will assign the result of the last returned Context to the _ (underscore) variable. In other words:

```
> "192.168.1.110"

192.168.1.110

> _ | ipv4_address

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
```

```
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>

> _ | nmap

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  +-80/tcp (HTTP) <via nmap_result_#1>
  +-22/tcp (SSH) <via nmap_result_#0>
  `-389/tcp (LDAP) <via nmap_result_#2>

> _ | nikto

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
    `-Found #10 Vulnerabilities <via nikto>

Vulnerabilities:
----------------

+-------------------------------------------------------------------------------------------+---------
| VULNERABILITY DESCRIPTION                                                                 | URL
+-------------------------------------------------------------------------------------------+---------
| Retrieved x-powered-by header: PHP/5.3.3-7+squeeze15                                      | http://1
+-------------------------------------------------------------------------------------------+---------
| Apache/2.2.16 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42  | http://1
| (final release) and 2.0.64 are also current.                                              |
+-------------------------------------------------------------------------------------------+---------
| DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en- | http://1
| us/library/e8z01xdh%28VS.80%29.aspx for details.                                          |
```

```
+-------------------------------------------------------------------------------------------+---------
| /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive    | http://1
| information via certain HTTP requests that contain specific QUERY strings.               |
+-------------------------------------------------------------------------------------------+---------
| /files/: Directory indexing found.                                                        | http://1
+-------------------------------------------------------------------------------------------+---------
| /files/: This might be interesting...                                                     | http://1
+-------------------------------------------------------------------------------------------+---------
| /img/: Directory indexing found.                                                          | http://1
+-------------------------------------------------------------------------------------------+---------
| /img/: This might be interesting...                                                       | http://1
+-------------------------------------------------------------------------------------------+---------
| /icons/: Directory indexing found.                                                        | http://1
+-------------------------------------------------------------------------------------------+---------
| /icons/README: Apache default file found.                                                 | http://1
+-------------------------------------------------------------------------------------------+---------
```

Is equal to:

```
> "192.168.1.110" | ipv4_address | nmap | nikto

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
    `-Found #10 Vulnerabilities <via nikto>

Vulnerabilities:
----------------

+-------------------------------------------------------------------------------------------+---------
| VULNERABILITY DESCRIPTION                                                                 | URL
+-------------------------------------------------------------------------------------------+---------
| Retrieved x-powered-by header: PHP/5.3.3-7+squeeze15                                      | http://1
+-------------------------------------------------------------------------------------------+---------
| Apache/2.2.16 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42  | http://1
| (final release) and 2.0.64 are also current.                                              |
+-------------------------------------------------------------------------------------------+---------
| DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en- | http://1
| us/library/e8z01xdh%28VS.80%29.aspx for details.                                          |
```

```
+-------------------------------------------------------------------------------------------+---------
| /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive    | http://1
| information via certain HTTP requests that contain specific QUERY strings.               |
+-------------------------------------------------------------------------------------------+---------
| /files/: Directory indexing found.                                                        | http://1
+-------------------------------------------------------------------------------------------+---------
| /files/: This might be interesting...                                                     | http://1
+-------------------------------------------------------------------------------------------+---------
| /img/: Directory indexing found.                                                          | http://1
+-------------------------------------------------------------------------------------------+---------
| /img/: This might be interesting...                                                       | http://1
+-------------------------------------------------------------------------------------------+---------
| /icons/: Directory indexing found.                                                        | http://1
+-------------------------------------------------------------------------------------------+---------
| /icons/README: Apache default file found.                                                 | http://1
+-------------------------------------------------------------------------------------------+---------
```

1.3.7 Conditional Expressions

Hackersh follows the Pythonect Control Flow Tools concept, and you can use the Context key/value pairs in a conditional expression:

```
> "192.168.1.110" | ipv4_address | nmap | _['PORT'] == '80'

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
```

The expression may contain any number of Python Boolean Operations:

```
> "192.168.1.110" | ipv4_address | nmap | _['PORT'] == '8080' or _['SERVICE'] == 'HTTP'

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
```

```
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
`-80/tcp (HTTP) <via nmap_result_#1>
```

As well as Python functions:

```
> "192.168.1.110" | ipv4_address | nmap | int(_['PORT']) < 1024

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  +-80/tcp (HTTP) <via nmap_result_#1>
  +-22/tcp (SSH) <via nmap_result_#0>
  `-389/tcp (LDAP) <via nmap_result_#2>
```

1.3.8 Map, Reduce, and Filter

Hackersh uses Pythonect's automatic parallelization feature. Whenever a Hackersh component returns more than one context, it would automatically map each Context to it's own thread.

Most Hackersh components are one-to-one or one-to-many. For example the `print` component is one-to-one. It will print the current context on the flow:

```
> "192.168.1.110" | ipv4_address | nmap | print
Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | SSH           |
+--------------+---------------+
| Port         | 22            |
+--------------+---------------+

Graph:
------
```

```
192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-22/tcp (SSH) <via nmap_result_#0>


Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>


Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | LDAP          |
+--------------+---------------+
| Port         | 389           |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-389/tcp (LDAP) <via nmap_result_#2>
```

Some Hackersh components are many-to-one. They will always end with `_all` postfix. For example the `print_all` component is many-to-one. It will reduce all contexts to one context and print it:

```
> "192.168.1.110" | ipv4_address | nmap | print_all

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  +-80/tcp (HTTP) <via nmap_result_#1>
  +-22/tcp (SSH) <via nmap_result_#0>
  `-389/tcp (LDAP) <via nmap_result_#2>
```

Note: When running Hackersh interactively (i.e. when commands are read from a tty) it will automatically reduce all the contexts and print the reduced context result.

To filter a reduced context you can use the / (div, forward slash) operator and a boolean expression (like *Conditional Expressions*):

```
> _/"SERVICE == 'HTTP' or SERVICE == 'HTTPS'"

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
```

Or:

```
> _/"PORT == '80' and PROTO == 'TCP'"

Properties:
-----------

+--------------+---------------+
| Property     | Value         |
+--------------+---------------+
| Port         | 80            |
+--------------+---------------+
| Ipv4_Address | 192.168.1.110 |
+--------------+---------------+
| Name         | 192.168.1.110 |
+--------------+---------------+
| Service      | HTTP          |
```

```
+--------------+---------------+
| Proto        | TCP           |
+--------------+---------------+

Graph:
------

192.168.1.110 <via str>
`-192.168.1.110 <via ipv4_address>
  `-80/tcp (HTTP) <via nmap_result_#1>
```

Note: As oppose to *Conditional Expressions*. This Query Language doesn't require `_[]` around context key names

---

1.3.9 Importing and Exporting Context

Hackersh lets you export and import your work at any time. You can use the `write` component to save a given context (or `write_all` to save all contexts) into file:

```
> "192.168.1.110" | ipv4_address | nmap | nikto | write 'web.json'
```

Reading it back is as easy as this:

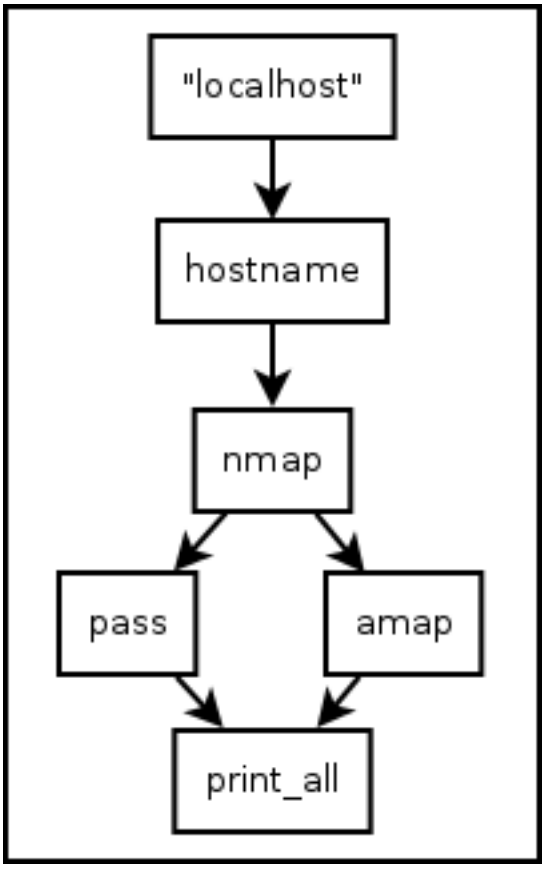
```
> read 'web.json' | print_all
```

The file format is determined by the extension you use in the file name (i.e., .json for a JSON file). Type `info read` or `info write` to see a complete list of supported file formats.

1.3.10 Writing and Running Scripts

Hackerh uses Pythonect as it's scripting engine. Pythonect provides both a visual programming language and a text-based scripting language.

The visual programming language is based on the idea of a diagram with "boxes and arrows":



Note: This is an export (PNG) of a diagram made in Dia. It's not actually a Hackersh script. The script is the actual **.dia** file.

Running a diagram is as easy as:

```
$ hackersh alternate_nmap_amap_scan.dia
```

The text-based scripting language (same syntax as used in the shell) aims to combine the quick and intuitive feel of shell scripting, with the power of Python. Open your favorite editor and type:

```
"localhost" -> hostname -> nmap -> [pass, amap] -> print_all
```

Save it as `alternate_nmap_amap_scan.hs` and run it as follows:

```
$ hackersh alternate_nmap_amap_scan.hs
```

Note: On BSD'ish Unix systems, Hackersh text-based scripts can be made directly executable, by putting the line (The #! must be the first two chars of the file):

```
#! /usr/bin/env hackersh
```

(assuming that Hackersh is on the user's PATH) at the beginning of the text-based script and giving the file an executable mode.

For more examples (in both, visual and text flavors) see the examples/ directory.

1.4 Development

Hackersh is under active development, and contributors are welcome.

If you have a feature request, suggestion, or bug report, please open a new issue on GitHub.

1.4.1 Contributor License Agreement

Before we can accept code, patches or pull requests on GitHub, there's a quick web form we need you to fill out here (**scroll to the bottom!**).

Hackersh's CLA is a copy of the one used by Sun Microsystems for all contributions to their projects.

This particular agreement has been used by other software projects in addition to Sun and is generally accepted as reasonable within the Open Source community.

More about CLAs

1.4.2 Source Control

Hackersh source is controlled with Git, the lean, mean, distributed source control machine.

The repository is publicly accessable.

```
git clone git://github.com/ikotler/hackersh.git
```

The project is hosted on GitHub.

Git Branch Structure

Feature / Hotfix / Release branches follow a Successful Git Branching Model. Git-flow is a great tool for managing the repository. I highly recommend it.

`develop` The "next release" branch. Likely unstable.

`master` Current production release (0.3) on PyPi.

Each release is tagged.

When submitting patches, please place your feature/change in its own branch prior to opening a pull reqeust on GitHub.

1.4.3 Adding New Components

TBD

1.4.4 Building the Docs

Documentation is written in the powerful, flexible, and standard Python documentation format, reStructured Text. Documentation builds are powered by the powerful Pocoo project, Sphinx. The *API Documentation* is mostly documented inline throughout the module.

The Docs live in `hackersh/doc`. In order to build them, you will first need to install Sphinx:

```
$ pip install sphinx
```

Then, to build an HTML version of the docs, simply run the following from the **doc** directory:

```
$ make html
```

Your `doc/_build/html` directory will then contain an HTML representation of the documentation, ready for publication on most web servers.

You can also generate the documentation in **epub**, **latex**, and **json**.

CHAPTER

TWO

API REFERENCE

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

Note: Until the first *stable* Hackersh version (1.0.0) is released, we reserve the right to break the API at any time.

2.1 `hackersh` — Parse and execute Hackersh code

This Python module provides the capability to parse and evaluate a string as Hackersh code

`hackersh.`**`parse`**(*source*)

Parse text-mode Hackersh scripting language into a directed graph (i.e. networkx.DiGraph)

Args: source: A string representing text-based Hackersh code.

Returns: A directed graph (i.e. networkx.DiGraph) of Hackersh symbols.

Raises: SyntaxError: An error occurred parsing the code.

`hackersh.`**`eval`**(*source*, *namespace*)

Evaluate Hackersh code in the context of locals.

Args: source: A string representing text-based Hackersh code or networkx.DiGraph instance. namespace: A dictionary with components.

Returns: The return value is the result of the evaluated code.

Raises: SyntaxError: An error occurred parsing the code.

CHAPTER

THREE

ADDITIONAL NOTES

Design notes, legal information and changelog are here for the interested.

3.1 Hackersh Changelog

Hackersh Changelog

HEAD

o Delete internal component "submit" for now. It will be back in the future, together with a nice set of HTTP components.

o Rename "iterate_links" to "web_crawler". For example: "http://localhost" -> url -> web_crawler -> ...

o All builtin shell commands and external components will display usage if *-h* or *–help* command line is passed. Example: nmap -h

o Introduce *clipboard* - new internal component for pasting text from Clipboard. Example: (Copy a URL) and then *clipboard | url | w3af*

o Introduce *regex_expand* - new internal component for inverting Regex. Example: "http://localhost/index[0-9].html" -> regex_expand -> url

o Root Components can be piped via __STDIN__ key in Context

o Add Components: *write*, *write_all*, and *read* to Write/Read Contexts. For more info see: *info write*, *info write_all*, and *info read*

o Hackersh loads Environment Variables on Load. For example: IPV4_ADDRESS="127.0.0.1" ./bin/hackersh -c '_ | nmap | print_all'

o Support Multi-Line in Interactive Console via Backslash ('). For example:

show all

Or: x = 5 And etc.

o *print* component will print a given context, while *print_all* will join (i.e. reduce) all contexts into a single tree and print that tree.

o Components can return Component-level Errors via HackershError() class. Aggregation occurs at Console-level prior to printing the return value.

o Hackersh Console Prompt can be customize via PROMPT variable. Example: PROMPT='% '

o Autocomplete with TAB. Examples: sho<TAB> info nm<TAB> "127.0.0.1" | <TAB> "127.0.0.1" | ipv4_<TAB> and etc.

o Add *show* Command (e.g. show)

o Add *info* Command (e.g. info nmap, info ipv4_address, and etc.)

o Add *help* Command (e.g. help help, help exit, and etc.)

o And Limited auto-ocomplete for Shell-like expressions (e.g. ./<TAB> and /<TAB>)

o Automatically Handle (i.e. return False) and Log Exceptions from Components. To see the a given Component Exceptions use debug=True

o Support realtime STDOUT and STDERR output via debug=True (e.g. "127.0.0.1" -> ipv4_address -> nmap(debug=True))

o Support Shell-style Arg Passing (i.e. "127.0.01" -> ipv4_address -> nmap -p22)

o Support "Smooth" Shell Pipeline Experience (i.e. /bin/cat /etc/hosts | /usr/bin/grep "127.0.0.1" | /usr/bin/awk '{ print $1 }' | /usr/bin/tr -d 'n' | ipv4_address | mmap)

o Change DEAFULT_QUERY and DEFAULT_FILTER to support the new Graph Search

Language. Example: 'context['PROTO'] == "TCP" and context['PORT'] == "80"'

Changed to: 'PROTO == "TCP" and PORT == "80"'

o Context can be searched via '/' (read: div) operator

Example: "127.0.0.1" -> ipv4_address -> nmap

Then: _/"PORT == '80"' -> nikto

Or: _/"PROTO == 'TCP"'/"PORT == '21"' -> ...

And etc.

o Component Entry Point is changed from run() to main(). Use run() for external inovcation (i.e. from The Shell, or Python Code) and main() for internal inovcation (i.e. from another Component). The run() method will call the main() method, but will perform init tasks before doing so.

o Context is now a Directed Graph

o Remove RemoteSessionContext, there's only one Context Object: class Context

o Delete _ordereddict.py

Hackersh 0.2 [2013-05-02]

o Introduce *dnsdict6* - new external component for: Information Gathering / Network Analysis / DNS Analysis dnsdict6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org Example: "hackersh.org" -> domain -> dnsdict6("-4 -s") -> ...

o Implement SimpleRegExHandler class - a new Pseudo SAX Content Handler class for processing output using regex

o amap, nikto, nmap, ping, w3af, xprobe2, and browse: Change DEFAULT_QUERY to evaluate context['IPV4_ADDRESS'] before context['HOSTNAME']

o Implement NbtScanStdoutOutputHandler class. Rewrite nbtscan to use it

o Implement SqlMapStdoutOutputHandler class. Rewrite sqlmap to use it

o Implement StdoutOutputHandler - a new Pseudo SAX Content Handler base class for stdout processing

o Implement ExternalComponentStreamOutput - a new base class for generic SAX-style output parsing. Change ExternalComponentStdoutOutput and ExternalComponentFileOutput to inherit from it.

o Implement shell_split() and replace shlex.split() with it. shell_split() will not remove double quotes (i.e. "") when splitting DEFAULT_QUERY.

o sqlmap: Change DEFAULT_QUERY to use "inline IF" to avoid:

TypeError: unsupported operand type(s) for +: 'bool' and 'str'

Whenever: context['COOKIES'] = False

o Introduce *ipv6_address* - new root component for processing IPv6 Address. Example: "::1" -> ipv6_address -> ...

o Introduce *domain* - new root component for processing domain names. Example: "hackersh.org" -> domain -> ...

o Implement HackershError Exception class and add 3 new error messages: XXX: not enough data to start (if Component Filter is False) XXX: command not found (if Ext. Component filename is missing) XXX: unable to parse (if all Output Handlers failed)

o Split hackersh/network.py and hackersh/misc.py into multiple files and implement a simple plug-in architecture to load them during startup

o Add support for BackTrack 5R3 and 5R2

Hackersh 0.1 [2013-04-01]

o Initial commit

3.2 License

Hackersh licensed under GPLv2+:

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
```

distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

```
  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License.  (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,
```

```
    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code.  (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it.  For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable.  However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License.  Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

  5. You are not required to accept this License, since you have not
signed it.  However, nothing else grants you permission to modify or
distribute the Program or its derivative works.  These actions are
prohibited by law if you do not accept this License.  Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions.  You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

  7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all.  For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
```

```
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices.  Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded.  In such case, this License incorporates
the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number.  If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation.  If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission.  For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this.  Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
```

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be

```
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  'Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```