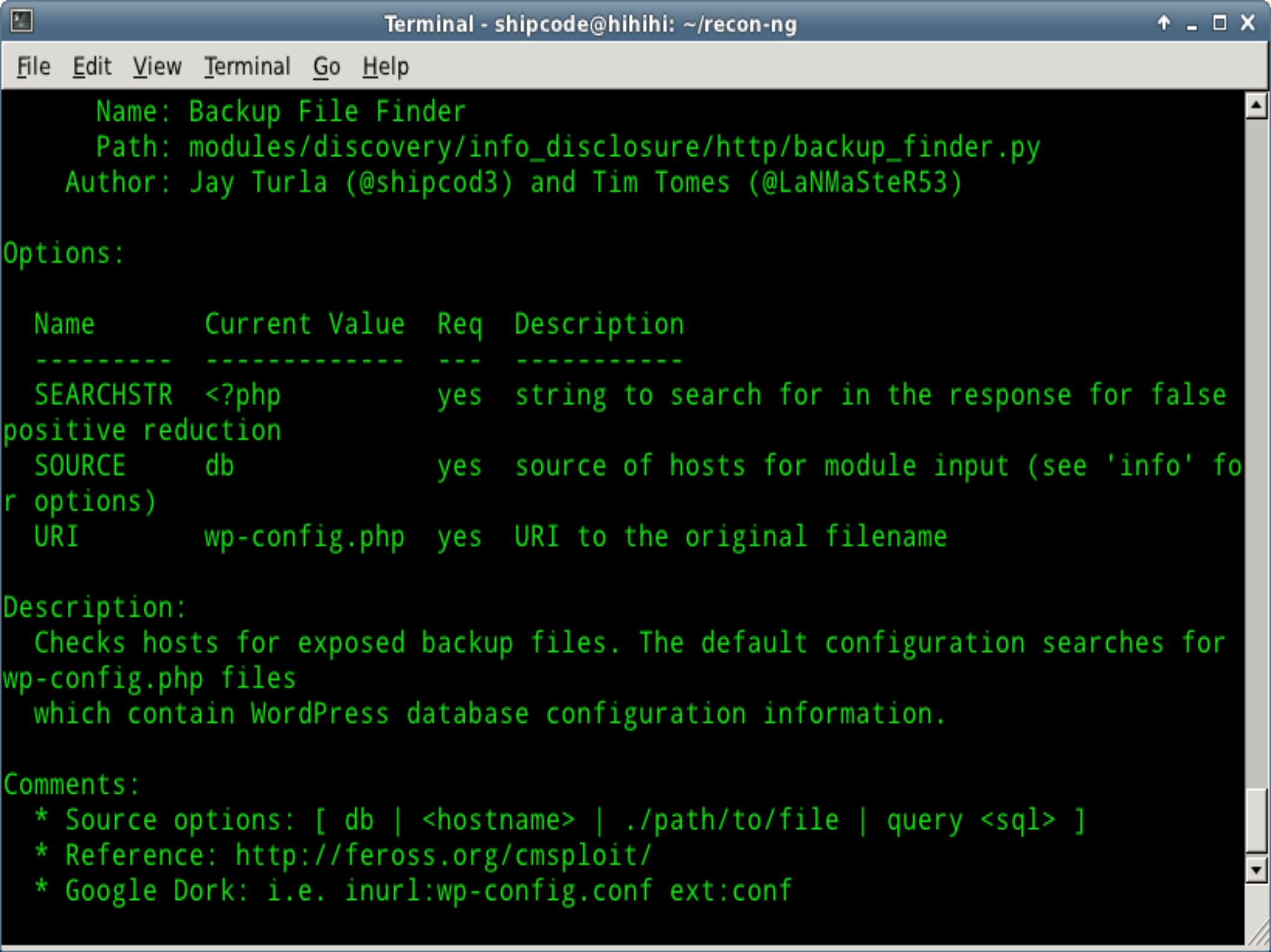
Diving into recon-ng

~ primarch victus

Who is primarch victus?

- **My name is Jay Turla**
- **I am one of the developers and contributors of the recon-ng framework**
- **I have been listed in the hall of fames of Adobe, Attack-Secure, Nokia, Microsoft, MailChimp, Constant Contract, Sprout Social, IntegraXor HMI, Puppet Labs, etc. for my responsible disclosures of security vulnerabilities.**
- **One of the recipients of Freelancer.com's Whitehat Badge for reporting 2 vulnerabilities to their security team.**
- **I work as an I.T. Lecturer at the two branches of Informatics Cebu and as a security researcher at Infosec Institute.**

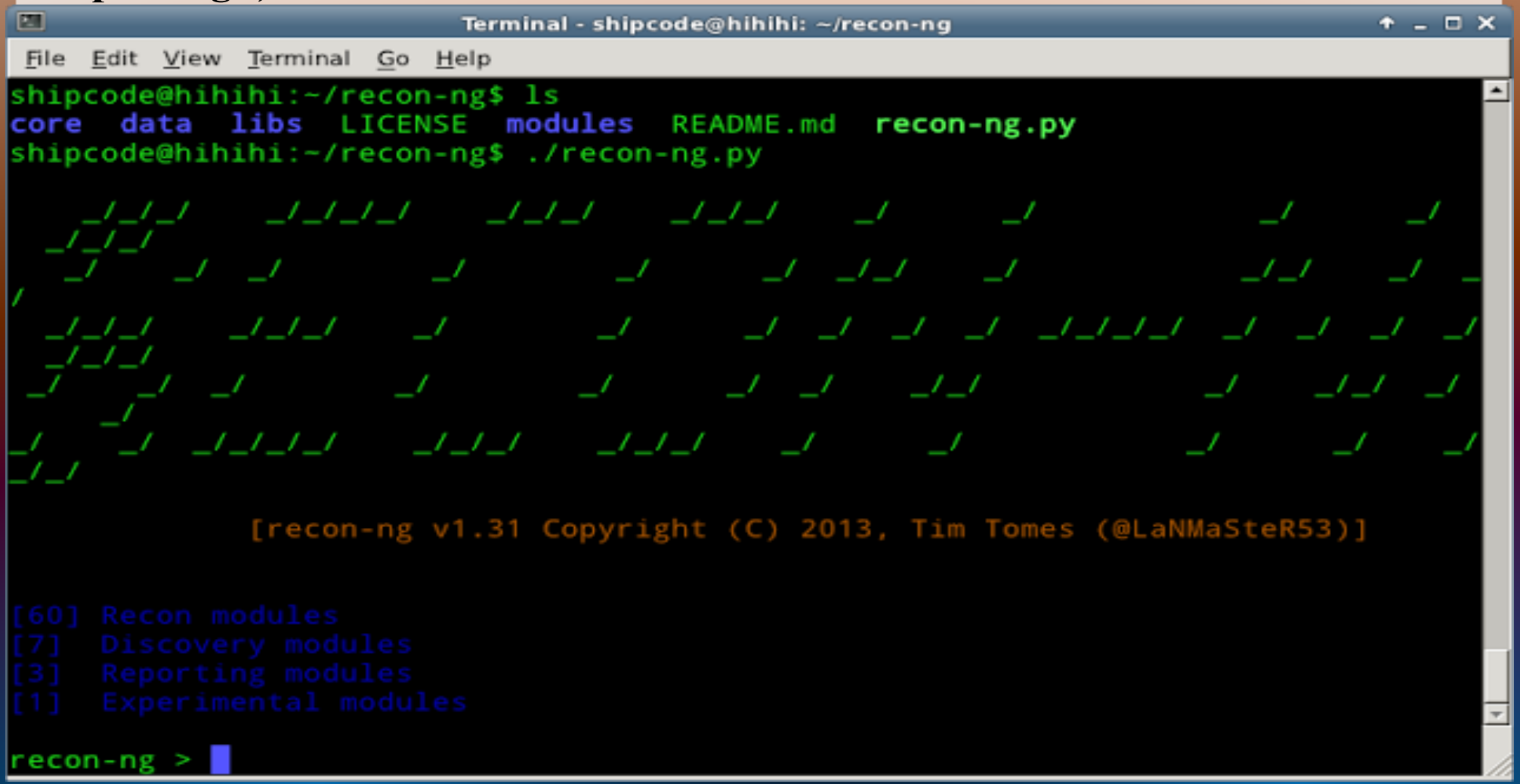

Terminal - shipcode@hihihi: ~/recon-ng
File Edit View Terminal Go Help
Name: Backup File Finder
Path: modules/discovery/info_disclosure/http/backup_finder.py
Author: Jay Turla (@shipcod3) and Tim Tomes (@LaNMaSteR53)
Options:
Name Current Value Req Description
SEARCHSTR <?php yes string to search for in the response for false positive reduction
SOURCE db yes source of hosts for module input (see 'info' for options)
URI wp-config.php yes URI to the original filename
Description:
Checks hosts for exposed backup files. The default configuration searches for wp-config.php files
which contain WordPress database configuration information.
Comments:
* Source options: [db | <hostname> | ./path/to/file | query <sql>]
* Reference: http://feross.org/cmsploit/
* Google Dork: i.e. inurl:wp-config.conf ext:conf

Introduction

- Recon-ng is an open-source framework coded in python by Tim Tomes a.k.a LaNMaSteR53.
- Its interface is modeled after the look of the Metasploit Framework but it is not for exploitation or for spawning a meterpreter session or a shell, it is for web-based reconnaissance and information gathering.
- It focuses Reconnaissance, Discovery, and Reporting which are steps 1, 2 and 4 of the Web Application Penetration Testing Methodology.

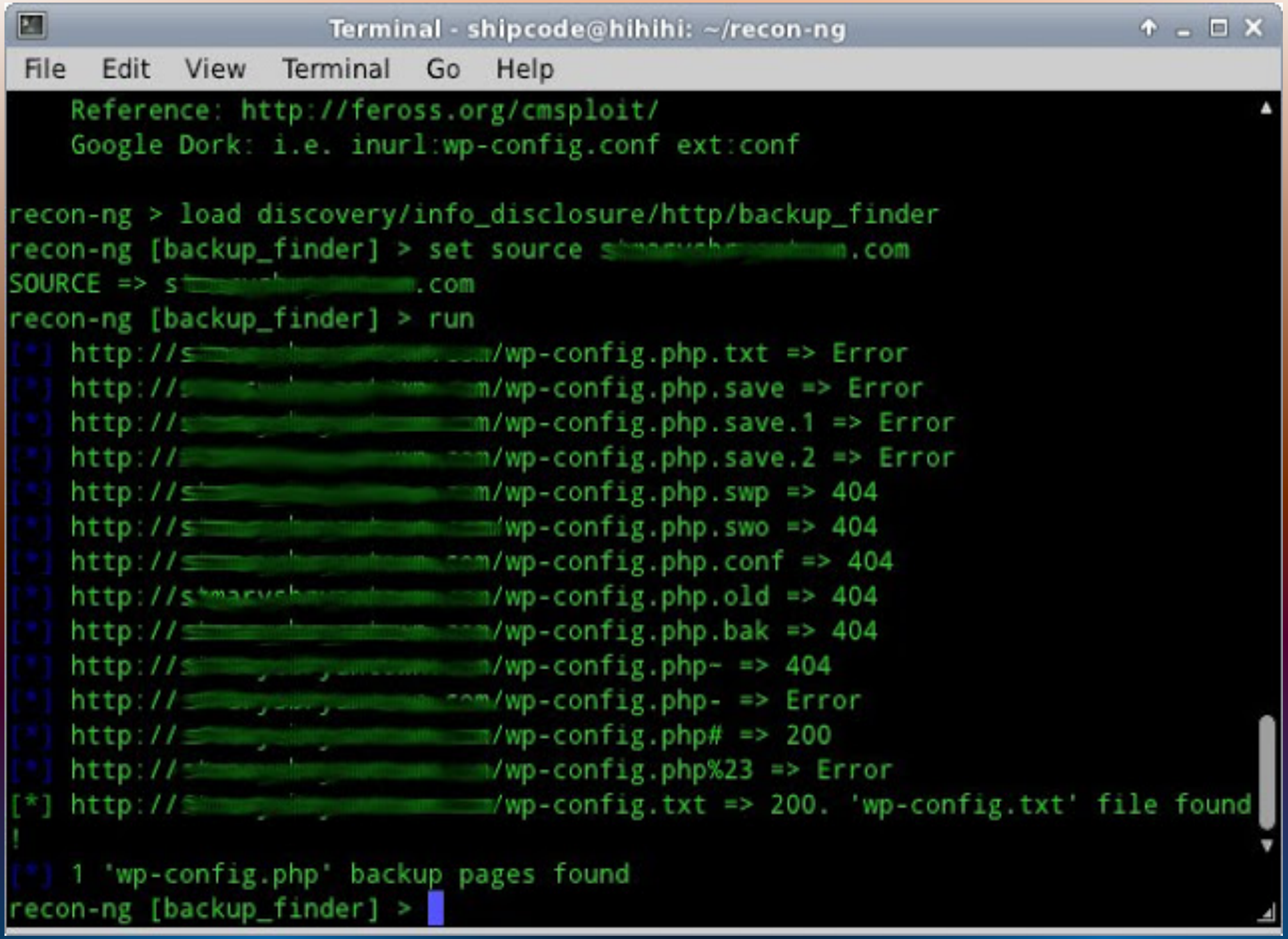
The Modules

- **Modules are categorized into Discovery, Experimental, Recon and Reporting :)**



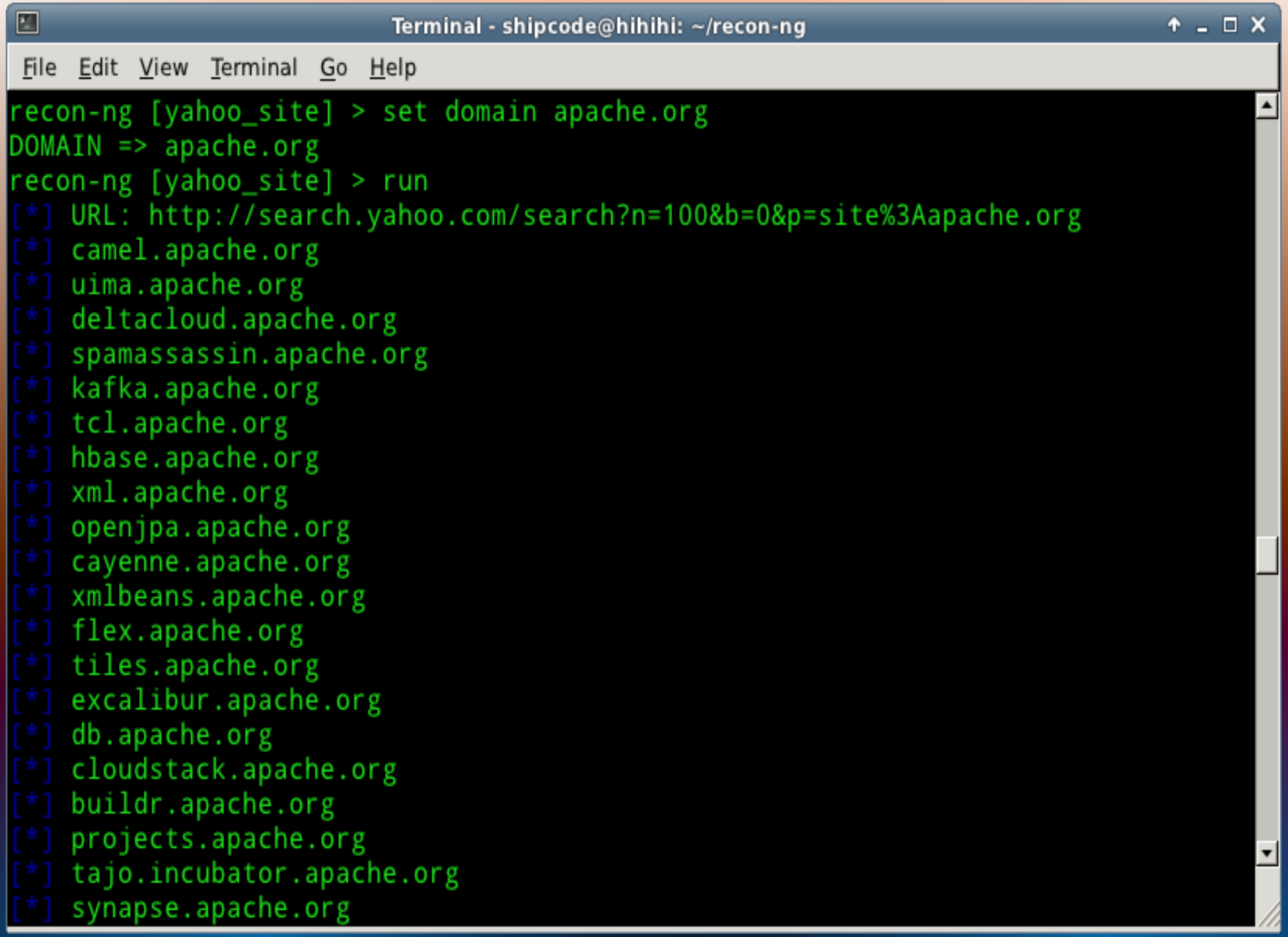
Discovery Modules

- Modules that can be used for finding exploitable files like file uploads, error logs, server statuses, php information, etc.
- Backup File Finder
- Examples: *Dot Net Nuke Remote File Upload Vulnerability Checker, GenericRestaurantMenu Vulnerability Page Finder and Validator, DNS Cache Snooper, Webwiz Rich Text Editor File Upload Page Finder*
- This is the category where I contributed mostly.


Terminal - shipcode@hihihi: ~/recon-ng
File Edit View Terminal Go Help
Reference: http://feross.org/cmsploit/
Google Dork: i.e. inurl:wp-config.conf ext:conf
recon-ng > load discovery/info_disclosure/http/backup_finder
recon-ng [backup_finder] > set source
.com
SOURCE => s
.com
recon-ng [backup_finder] > run
[*] http://
/wp-config.php.txt => Error
[*] http://
/wp-config.php.save => Error
[*] http://
/wp-config.php.save.1 => Error
[*] http://
/wp-config.php.save.2 => Error
[*] http://
/wp-config.php.swp => 404
[*] http://
/wp-config.php.swo => 404
[*] http://
/wp-config.php.conf => 404
[*] http://
/wp-config.php.old => 404
[*] http://
/wp-config.php.bak => 404
[*] http://
/wp-config.php~ => 404
[*] http://
/wp-config.php- => Error
[*] http://
/wp-config.php# => 200
[*] http://
/wp-config.php%23 => Error
[*] http://
/wp-config.txt => 200. 'wp-config.txt' file found
!
[*] 1 'wp-config.php' backup pages found
recon-ng [backup_finder] >

The Recon Modules

- Used for domain lookups, dns lookups, mail host lookups, enumerating hostnames, enumerating subdomains, enumerating company emails, etc.
- Modules that leverages API's and Online Scanners
- Majority of the modules for recon-ng are categorized here
- Examples: *Flickr Geolocation Search, My-IP-Neighbors Lookup, McAfee Domain DNS Lookup, Yahoo Hostname Enumerator, Twitter Handles, punkSPIDER Vulnerabilty Finder*


Terminal - shipcode@hihihi: ~/recon-ng
File Edit View Terminal Go Help
recon-ng [yahoo_site] > set domain apache.org
DOMAIN => apache.org
recon-ng [yahoo_site] > run
[*] URL: http://search.yahoo.com/search?n=100&b=0&p=site%3Aapache.org
[*] camel.apache.org
[*] uima.apache.org
[*] deltacloud.apache.org
[*] spamassassin.apache.org
[*] kafka.apache.org
[*] tcl.apache.org
[*] hbase.apache.org
[*] xml.apache.org
[*] openjpa.apache.org
[*] cayenne.apache.org
[*] xmlbeans.apache.org
[*] flex.apache.org
[*] tiles.apache.org
[*] excalibur.apache.org
[*] db.apache.org
[*] cloudstack.apache.org
[*] buildr.apache.org
[*] projects.apache.org
[*] tajo.incubator.apache.org
[*] synapse.apache.org

Reporting Modules

- Used for creating a CSV or an HTML file containing the specified harvested data types.
- Modules: CSV File Creator, HTML Report Generator, List Creator, and PushPin Report Generator
- The Pushpin Report Generator creates a media and map HTML report for all of the PushPin data stored in the database.

file:///home/shipcode/results.html

Recon-ng Reconnaissance Report

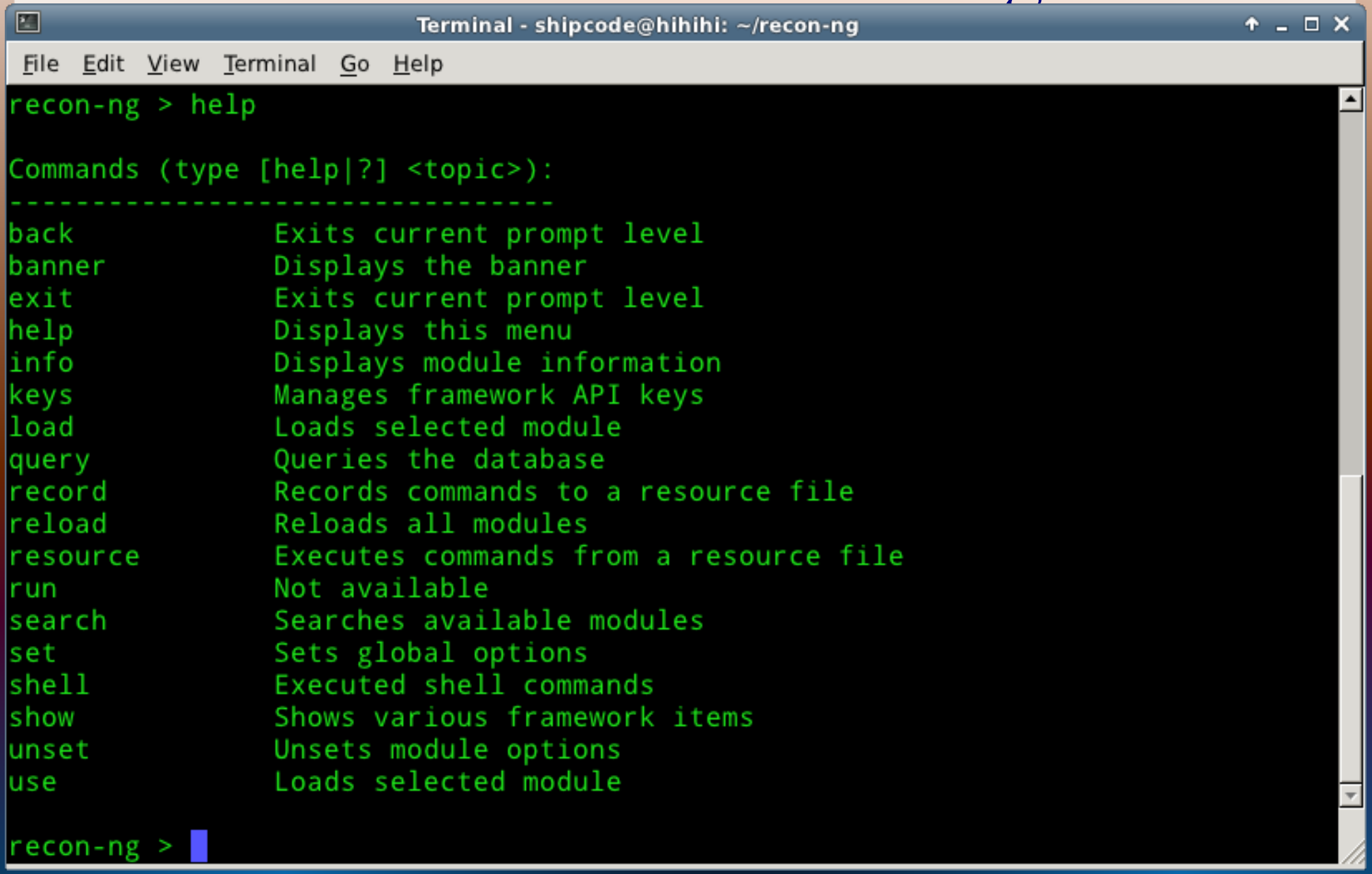
CONTACTS

First Name	Last Name	Email/Username	Title
Aaron	Bell		Specialist Field Service
Aaron	Bernstein		Deputy General Counsel and Vice President of Intel
Abdon	Badillo		Software Tech
Agus	Mulia		Programmer/Developer
Alessandra	Vega		Assistant Office Manager
Alper	Turken		Account Manager Networks
Amarildo	Veira		Principal Systems Engineer at Motorola Home and Ne
Amit	Bhavnani		Senior Test Engineer
Amy	Haukeness		Technical Solutions
Andrej	Koperdan		Manager Corporate Development and Strategic Transa
Anne	Pearce		Section Manager
Anthony	Williams		Sales Manager
Anthony	Braskich		Senior Research Engineer
Anthony	Schooler		Engineer
Anthony	Benckenga		Manager/Supervisor
Ariana	Gonzalez		Engineer Firmware
Art	Psyhogoes		Sales Support Representative
Bert	Van Der Zaag		iDEN UI Design Manager, Design Integration
Bill	Williams		Operations Manager
Bill	Brewer		Global Information Technology Configuration Manage
Bob	Logalbo		PRINCIPLE Staff Engineer
Bob	Uskali		Engineer
Bob	Wallace		Integrated Supply Chain Law Department
Boris	Bekkerman		Principal Staff Engineer
Brian	Bauerschmidt		PRINCIPLE Staff Engineer
Brian	Carroll		Engineer Electronics Staff Principal
Brian	Gola		Engineer Test Senior
Carmen	Damelio		Manager Supply Management II
Catherine	Cai		Personal Communications Sector
Charles	Evans		Manager Project
Charles	Adams		Analyst Planning Senior

How You Can Help?

- Prerequisites: Git and Python
- Clone https://bitbucket.org/LaNMaSteR53/recon-ng
- Start playing with the framework
- Think of a module that can be useful and start coding one
- Push it!

Basic Framework Usage



Source Code Previews of Some Modules

```
def module_run(self):
    hosts = self.get_source(self.options['source']['value'], 'SELECT DISTINCT host FROM hosts WHERE host IS NOT NULL ORDER BY host')
    uri = self.options['uri']['value']
    searchstr = self.options['searchstr']['value']

    protocols = ['http', 'https']

    # some files are inspired by cmsploit
    uris = [uri, uri[:uri.rindex('.')]]
    exts = ['.txt', '.save', '.save.1', '.save.2', '.swp', '.swo', '.conf', '.old', '.bak', '~', '-', '#', '%23']
    filenames = []
    # mangle root uris to create a list of possible backup filenames
    for rooturi in uris:
        for ext in exts:
            filenames.append('%s%s' % (rooturi, ext))

    cnt = 0
    for host in hosts:
        flag = 0
        for proto in protocols:
            for filename in filenames:
                url = '%s://%s/%s' % (proto, host, filename)
                try:
                    resp = self.request(url, redirect=False)
                    code = resp.status_code
                except KeyboardInterrupt:
                    raise KeyboardInterrupt
                except:
                    code = 'Error'
                if code == 200 and searchstr in resp.text:
                    self.alert('%s => %s. \'%s\' file found!' % (url, code, filename))
                    cnt += 1
                    flag = 1
                    break
                else:
                    self.verbose('%s => %s' % (url, code))
            if flag: break
    self.output('%d \'%s\' backup pages found' % (cnt, uri))
```

```
def module_run(self):
    hosts = self.get_source(self.options['source']['value'], 'SELECT DISTINCT host FROM hosts WHERE host IS NOT NULL ORDER BY host')
    validate = self.options['validate']['value']

    # check all hosts for GenericRestaurantMenu Menu Categories Editor Page, SQL Query Info Disclosure, and Possible SQLi Vulnerbility
    protocols = ['http', 'https']
    cnt = 0
    for host in hosts:
        for proto in protocols:
            url = '%s://%s/Menu/admin/' % (proto, host)
            try:
                resp = self.request(url, redirect=False)
                code = resp.status_code
            except KeyboardInterrupt:
                raise KeyboardInterrupt
            except:
                code = 'Error'
            if code == 200 and 'Menu Categories' in resp.text:
                self.alert('%s => %s. Menu Categories Editor Page Found!' % (url, code))
                cnt += 1
                if validate:
                    vulncode = "%s://%s/menu/view.cfm?category_ID=1'" % (proto, host)
                    try:
                        resp = self.request(vulncode, redirect=False)
                        code = resp.status_code
                    except KeyboardInterrupt:
                        raise KeyboardInterrupt
                    except:
                        code = 'Error'
                    if code == 500 and 'Executing Database Query' in resp.text:
                        self.alert('%s => %s. SQL Query Info Disclosure and Possible SQLi (boolean-based blind) Vulnerability Found!' %
                        cnt += 1
                    else:
                        self.verbose('%s => %s' % (vulncode, code))
            else:
                self.verbose('%s => %s' % (url, code))
    self.output('%d possibly vulnerable pages found!' % (cnt))
```

```
class Module(framework.module):

    def __init__(self, params):
        framework.module.__init__(self, params)
        self.register_option('source', 'db', 'yes', 'source of hosts for module input (see \'info\' for options)')
        self.info = {
                    'Name': 'WhatWeb Web Technologies scan',
                    'Author': 'thrapt (thrapt@gmail.com) and Tim Tomes (@LaNMaSteR53)',
                    'Description': 'Leverages WhatWeb.net to determine the web technologies in use on the given host(s).',
                    'Comments': [
                                 'Source options: [ db | <hostname> | ./path/to/file | query <sql> ]'
                                ]
                    }

    def module_run(self):
        # handle sources
        hosts = self.get_source(self.options['source']['value'], 'SELECT DISTINCT host FROM hosts WHERE host IS NOT NULL ORDER BY host')

        url = 'http://whatweb.net/whatweb.php'
        for host in hosts:
            payload = {'target': host, 'format': 'json' }
            resp = self.request(url, method='POST', payload=payload)

            # parse returned json objects
            jsonobj = resp.json
            if jsonobj == None and resp.text:
                jsonobjs = [json.loads(x) for x in resp.text.strip().split('\n')]
            else:
                jsonobjs = [jsonobj]

            # output data
            for jsonobj in jsonobjs:
                tdata = [['Plugin', 'String'],['Target', jsonobj['target']]]
                for plugin in jsonobj['plugins']:
                    if 'string' in jsonobj['plugins'][plugin]:
                        value = ', '.join(jsonobj['plugins'][plugin]['string'])
                        tdata.append([plugin, value])
                if tdata: self.table(tdata, header=True)
```

DEMO!

-enuff with some talk sh***