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 fame 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.
Name: Backup File Finder
Path: modules/discovery/info_disclosure/http/backup_finder.py
Author: Jay Turla (@shipcod3) and Tim Tomes (@LaNMaSteR53)

Options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Current Value</th>
<th>Req</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCHSTR</td>
<td>&lt;?php</td>
<td>yes</td>
<td>string to search for in the response for false positive reduction</td>
</tr>
<tr>
<td>SOURCE</td>
<td>db</td>
<td>yes</td>
<td>source of hosts for module input (see 'info' for options)</td>
</tr>
<tr>
<td>URI</td>
<td>wp-config.php</td>
<td>yes</td>
<td>URI to the original filename</td>
</tr>
</tbody>
</table>

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


- This is the category where I contributed mostly.
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 stmaryshrm.com
SOURCE => stmaryshrm.com
recon-ng [backup_finder] > run

[*] http://stmaryshrm.com/wp-config.php.txt => Error
[*] http://stmaryshrm.com/wp-config.php.save => Error
[*] http://stmaryshrm.com/wp-config.php.save.1 => Error
[*] http://stmaryshrm.com/wp-config.php.save.2 => Error
[*] http://stmaryshrm.com/wp-config.php.swp => 404
[*] http://stmaryshrm.com/wp-config.php.swo => 404
[*] http://stmaryshrm.com/wp-config.php.conf => 404
[*] http://stmaryshrm.com/wp-config.php.old => 404
[*] http://stmaryshrm.com/wp-config.php.bak => 404
[*] http://stmaryshrm.com/wp-config.php- => 404
[*] http://stmaryshrm.com/wp-config.php- => Error
[*] http://stmaryshrm.com/wp-config.php# => 200
[*] http://stmaryshrm.com/wp-config.php%23 => Error
[*] http://stmaryshrm.com/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 Vulnerability Finder
recon-ng [yahoo_site] > set domain apache.org
DOMAIN => apache.org
recon-ng [yahoo_site] > run
[*] URL: http://search.yahoo.com/search?q=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.
- The Pushpin Report Generator creates a media and map HTML report for all of the PushPin data stored in the database.
# Recon-ng Reconnaissance Report

## CONTACTS

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Email/Username</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>Bell</td>
<td></td>
<td>Specialist Field Service</td>
</tr>
<tr>
<td>Aaron</td>
<td>Bernstein</td>
<td></td>
<td>Deputy General Counsel and Vice President of Intel</td>
</tr>
<tr>
<td>Abdon</td>
<td>Badillo</td>
<td></td>
<td>Software Tech</td>
</tr>
<tr>
<td>Agus</td>
<td>Mulla</td>
<td></td>
<td>Programmer/Developer</td>
</tr>
<tr>
<td>Alessandra</td>
<td>Vega</td>
<td></td>
<td>Assistant Office Manager</td>
</tr>
<tr>
<td>Alper</td>
<td>Turkmen</td>
<td></td>
<td>Account Manager Networks</td>
</tr>
<tr>
<td>Amarildo</td>
<td>Vieira</td>
<td></td>
<td>Principal Systems Engineer at Motorola Home and Ne</td>
</tr>
<tr>
<td>Amit</td>
<td>Bhavnani</td>
<td></td>
<td>Senior Test Engineer</td>
</tr>
<tr>
<td>Amy</td>
<td>Haukeness</td>
<td></td>
<td>Technical Solutions</td>
</tr>
<tr>
<td>Andrej</td>
<td>Koperdan</td>
<td></td>
<td>Manager Corporate Development and Strategic Transa</td>
</tr>
<tr>
<td>Anne</td>
<td>Pearce</td>
<td></td>
<td>Section Manager</td>
</tr>
<tr>
<td>Anthony</td>
<td>Williams</td>
<td></td>
<td>Sales Manager</td>
</tr>
<tr>
<td>Anthony</td>
<td>Braschick</td>
<td></td>
<td>Senior Research Engineer</td>
</tr>
<tr>
<td>Anthony</td>
<td>Schooler</td>
<td></td>
<td>Engineer</td>
</tr>
<tr>
<td>Anthony</td>
<td>Benchvenga</td>
<td></td>
<td>Manager/Supervisor</td>
</tr>
<tr>
<td>Ariana</td>
<td>Gonzalez</td>
<td></td>
<td>Engineer Firmware</td>
</tr>
<tr>
<td>Art</td>
<td>Psyhogoes</td>
<td></td>
<td>Sales Support Representative</td>
</tr>
<tr>
<td>Bert</td>
<td>Van Der Zaag</td>
<td></td>
<td>IDEN UI Design Manager, Design Integration</td>
</tr>
<tr>
<td>Bill</td>
<td>Williams</td>
<td></td>
<td>Operations Manager</td>
</tr>
<tr>
<td>Bill</td>
<td>Brewer</td>
<td></td>
<td>Global Information Technology Configuration Manage</td>
</tr>
<tr>
<td>Bob</td>
<td>Logalbo</td>
<td></td>
<td>PRINCIPLE Staff Engineer</td>
</tr>
<tr>
<td>Bob</td>
<td>Utzki</td>
<td></td>
<td>Engineer</td>
</tr>
<tr>
<td>Bob</td>
<td>Wallace</td>
<td></td>
<td>Integrated Supply Chain Law Department</td>
</tr>
<tr>
<td>Boris</td>
<td>Bekkerman</td>
<td></td>
<td>Principal Staff Engineer</td>
</tr>
<tr>
<td>Brian</td>
<td>Bauernschmidt</td>
<td></td>
<td>PRINCIPLE Staff Engineer</td>
</tr>
<tr>
<td>Brian</td>
<td>Carroll</td>
<td></td>
<td>Engineer Electronics Staff Principal</td>
</tr>
<tr>
<td>Brian</td>
<td>Gola</td>
<td></td>
<td>Engineer Test Senior</td>
</tr>
<tr>
<td>Carmen</td>
<td>Damello</td>
<td></td>
<td>Manager Supply Management II</td>
</tr>
<tr>
<td>Catherine</td>
<td>Cai</td>
<td></td>
<td>Personal Communications Sector</td>
</tr>
<tr>
<td>Charles</td>
<td>Evans</td>
<td></td>
<td>Manager Project</td>
</tr>
<tr>
<td>Charlie</td>
<td>Artemoto</td>
<td></td>
<td>Basket Division Senior</td>
</tr>
</tbody>
</table>
How You Can Help?

- Prerequisites: Git and Python
- Clone [https://bitbucket.org/LaNMaSteR53/recon-ng](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

recon-ng > help

Commands (type [help|?] <topic>):

- back: Exits current prompt level
- banner: Displays the banner
- exit: Exits current prompt level
- help: Displays this menu
- info: Displays module information
- keys: Manages framework API keys
- load: Loads selected module
- query: Queries the database
- record: Records commands to a resource file
- reload: Reloads all modules
- resource: Executes commands from a resource file
- run: Not available
- search: Searches available modules
- set: Sets global options
- shell: Executed shell commands
- show: Shows various framework items
- unset: Unsets module options
- use: Loads selected module

recon-ng >
Source Code Previews of Some Modules
def module_run(self):
    hosts = self.get_source(self.options['source']�이 'value']에서 'SELECT DISTINCT host FROM hosts WHERE host IS NOT NULL ORDER BY host')
    uri = self.options['uri']에서 'value'
    searchstr = self.options['searchstr']에서 'value'

    protocols = ['http', 'https]

    # some files are inspired by cmsploit
    uris = [uri, uri[:uri.reverse('.')]]
    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 Vulnerability
    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
            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
                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!' % (url, code))
                    cnt += 1
            else:
                self.verbose('%s => %s' % (vulncode, 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('
')]
        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***