



HUNT - Data Driven Web Hacking & Manual Testing

JP Villanueva

Trust & Security Engineer @bugcrowd
@swagnetow

Contributions

Motley crew @bugcrowd

SecEng and SecOps teams

Bug Hunters, Pentesters, Code Analysis

Burp Suite fans

Github contributors

The Problems

1. Increasingly large and complicated web applications that need manual testing
2. Applications assessment training lacks “tribal knowledge” of vulnerability location
3. No in-tool workflow for web hacking methodologies

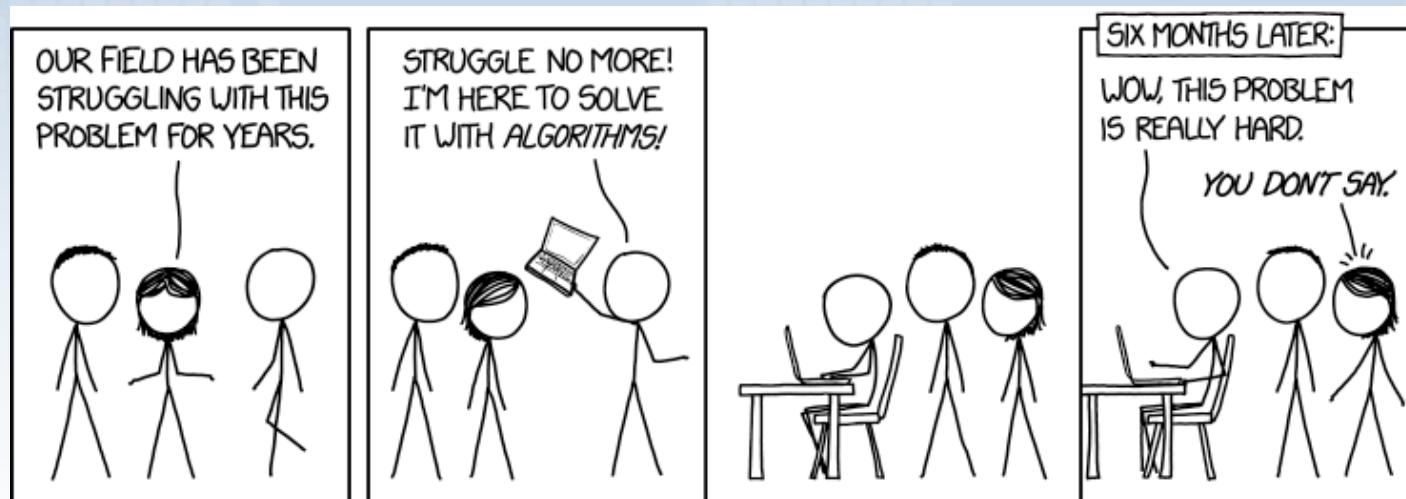
Current Solutions

1. Hacker who can eyeball and effectively find security bugs
 - a. May or may not have a methodology
 - b. Definitely has accrued “tribal knowledge”
 - c. Bug hunts and/or does consultant work
2. Dynamic Scanner
 - a. Limited test cases (fuzzing)
 - b. Cost prohibitive
 - c. Limited in detection cases (dynamic pages, errors, etc)
 - d. Complex sites are hard (auth)

Introducing HUNT

1. Tribal knowledge passive alerts
2. Methodology in Burp
3. Manual testing references in Burp

Level 1 - HUNT Scanner



Source: xkcd

Bug Location (Tribal Knowledge)

- Data from over 600+ bug bounty programs
 - ~2 web targets per program on average
 - 2017.appsecusa.org, 2016.appsecusa.org
 - ~15 parameters per target on average

⇒ $600 * 2 * 15 \approx 18,000$ parameters seen

Vulnerability Locations

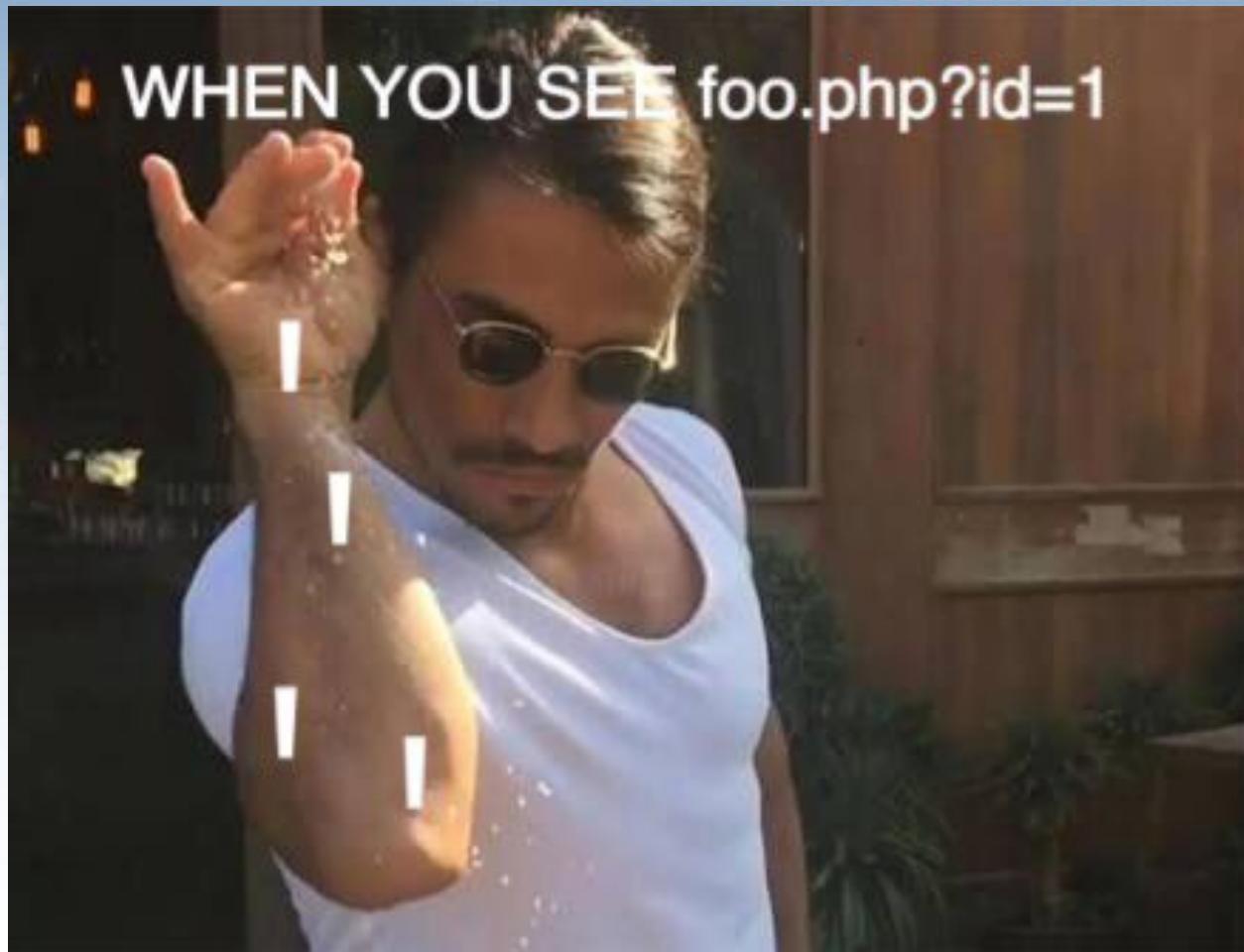
- ⇒ 18,000 parameters
 - Reduce to params with vulns on them
 - Reduce to only Critical and High severity bugs/vulns
 - Sort by recurring instances
 - Include top 5-10 reoccurring instances per vuln/bug category
 - Review top 100 for possible permutations manually and/or with regex
 - Manually add ancillary data (pentest/fuzzdb/seclists/etc)

Exhibit A

`https://2017.appsecusa.org/register?id=a`

The diagram illustrates the structure of the URL `https://2017.appsecusa.org/register?id=a`. It is divided into five segments by blue curved arrows:

- protocol: `https`
- subdomain: `2017`
- domain: `appsecusa.org`
- file or resource: `register`
- parameter and parameter value: `?id=a`



Source: Twitter

Alerts

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts HUNT - Scanner HUNT - Methodology

Vulnerability Classes

- Insecure Direct Object Reference (1)
- Server Side Request Forgery
- Debug & Logic Parameters
- Server Side Template Injection (1)
- OS Command Injection
- SQL Injection (1)
 - id (1)**
 - select
 - report
 - role
 - update
 - query
 - user
 - name
 - sort
 - where
 - search
 - params
 - process
 - row
 - view
 - table
 - from
 - sel
 - results
 - sleep
 - fetch
 - order
 - keyword
 - column
 - field
 - delete
 - string
 - number
 - filter
- File Inclusion & Path Traversal

Checked	Parameter	Host	Path
<input checked="" type="checkbox"/>	id	2017.appsecusa.org	/

Advisory Request Response Load/Save Results

Raw Headers Hex HTML Render

HTTP/1.1 200 OK
Date: Fri, 22 Sep 2017 02:45:38 GMT
Server: Apache/2.4.18 (Ubuntu)
Strict-Transport-Security: max-age=15768000
Set-Cookie: wfvt_2244512973=59c47952bc101; expires=Fri, 22-Sep-2017 03:15:38 GMT; Max-Age=1800; path=/; secure; HttpOnly
Link: <https://2017.appsecusa.org/wp-json/>; rel="https://api.w.org/"
Link: <https://2017.appsecusa.org/>; rel=shortlink
Vary: Accept-Encoding
Content-Length: 38710
Connection: close
Content-Type: text/html; charset=UTF-8

```
<!DOCTYPE html>
<html lang="en-US">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="user-scalable=no, width=device-width, initial-scale=1, maximum-scale=1">
```

?

< + > Type a search term

0 matches



OWASP
Open Web Application
Security Project

Advisory

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts HUNT – Scanner HUNT – Methodology

Vulnerability Classes

- Insecure Direct Object Reference (1)
- Server Side Request Forgery
- Debug & Logic Parameters
- Server Side Template Injection (1)
- OS Command Injection
- SQL Injection (1)
 - id (1)**
 - select
 - report
 - role
 - update
 - query
 - user
 - name
 - sort
 - where
 - search
 - params
 - process
 - row
 - view
 - table
 - from
 - sel
 - results
 - sleep
 - fetch
 - order
 - keyword
 - column
 - field
 - delete
 - string
 - number
 - filter
- File Inclusion & Path Traversal

Checked	Parameter	Host	Path
<input type="checkbox"/>	id	2017.appsecusa.org	/

Advisory Request Response Load/Save Results

Location: <https://2017.appsecusa.org/>

HUNT located the **id** parameter inside of your application traffic. The **id** parameter is most often susceptible to SQL Injection. HUNT recommends further manual analysis of the parameter in question.

For SQL Injection HUNT references The Bug Hunters Methodology SQL Injection references table:

[PentestMonkey's MySQL Injection Cheat Sheet](#)
[Reiner's MySQL Injection Filter Evasion](#)
[EvilSQL's Error/Union/Blind MSSQL Cheat Sheet](#)
[PentestMonkey's MSSQL SQL Injection Cheat Sheet](#)
[PentestMonkey's Oracle SQL Cheat Sheet](#)
[PentestMonkey's PostgreSQL Cheat Sheet](#)
[Access SQL Injection Cheat Sheet](#)
[Access SQL Injection Cheat Sheet](#)
[PentestMonkey's Ingres SQL Injection Cheat Sheet](#)
[PentestMonkey's DB2 SQL Injection Cheat Sheet](#)
[PentestMonkey's Informix SQL Injection Cheat Sheet](#)
[SQLite3 Injection Cheat Sheet](#)
[Ruby on Rails \(ActiveRecord\) SQL Injection Guide](#)



OWASP
Open Web Application
Security Project

Bug Location by Bug/Vuln Class



Source: Shirtoid.com

SQL Injection

{regex + perm} id	{regex} select	{regex} report	{regex} role
{regex} update	{regex} query	{regex + perm} user	{regex + perm} name
{regex} sort	{regex} where	{regex + perm} search	{regex} params
{regex} process	{regex + perm} row	{regex + perm} view	{regex} table
{regex + perm} from	{regex + perm} sel	{regex} results	{regex} sleep
{regex} fetch	{regex + perm} order	{regex} keyword	{regex} count
{regex + perm} column	{regex} input	{regex + perm} key	
{regex + perm} code	{regex + perm} field	{regex} delete	{type} Custom headers
{regex} string	{regex} number	{regex + perm} filter	{type} JSON and XML services

File Includes/Directory Indexing

{regex + perm} file	{regex} location	{regex} locale	{regex + perm} path
{regex} display	{regex} load	{regex + perm} read	{regex} retrieve
{regex + perm} folder	{regex} style	{regex + perm} doc	{regex} document
{regex} root	{regex} pdf	{regex} pg	{regex} include
{regex} list	{regex} view	{regex} img	{regex} image

Server Side Request Forgery 🔥🔥🔥

{regex + perm} dest	{regex} redirect	{regex + perm} uri	{regex} path
{regex} continue	{regex + perm} url	{regex} window	{regex} next
{regex} data	{regex} reference	{regex + perm} site	{regex} html
{regex + perm} val	{regex} validate	{regex} domain	{regex} callback
{regex} return	{regex + perm} page	{regex} feed	{regex} host
{regex} port			

OS Command Injection

{regex} daemon	{regex + perm} upload	{regex + perm} dir
{regex} execute	{regex + perm} download	{regex + perm} log
{type} .cgi	{regex} ip	
{regex} cli		

Insecure Direct Object Reference

{regex + perm} id	{regex + perm} user	
{regex + perm} account	{regex + perm} number	
{regex + perm} order	{regex + perm} no	
{regex + perm} doc	{regex + perm} key	
{regex + perm} email	{regex + perm} group	
{regex + perm} profile	{regex + perm} edit	REST numeric paths

Server Side Template Injection

{regex + perm} template	content	id
preview	redirect	view
activity	name	

Debug & Logic Parameters

access	admin	dbg
debug	edit	grant
test	alter	clone
create	delete	disable
enable	exec	execute
load	make	modify
rename	reset	shell
toggle	adm	root
cfg	config	

HUNT Scanner Implementation

```
IBurpExtender, IExtensionStateListener, IScannerCheck, ITab, ITextEditor

def doPassiveScan(self, request_response):
    raw_request = request_response.getRequest()
    raw_response = request_response.getResponse()
    request = self.helpers.analyzeRequest(raw_request)
    response = self.helpers.analyzeResponse(raw_response)

    parameters = request.getParameters()
    url = self.helpers.analyzeRequest(request_response).getUrl()
    vuln_parameters = self.issues.check_parameters(self.helpers, parameters)

    is_not_empty = len(vuln_parameters) > 0

    if is_not_empty:
        self.issues.create_scanner_issues(self.view, self.callbacks, self.helpers, vuln_parameters, request_response)

    # Do not show any Bugcrowd found issues in the Scanner window
    return []
```

Live Demo: PLEASE WORK PLEASE

DEMO GODS



**PLEASE MAKE THIS DEMO
WORK**

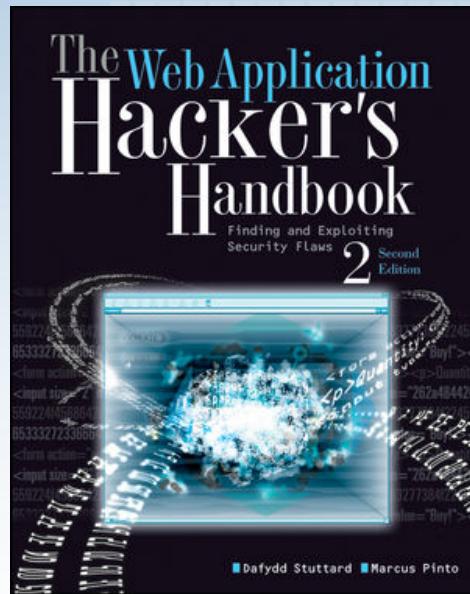
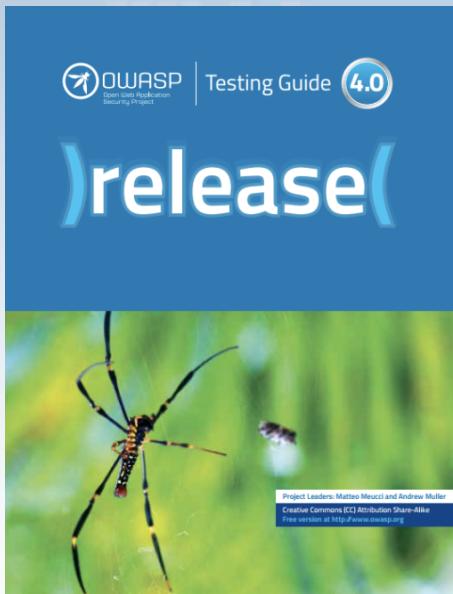
memegenerator.net

Level 2 - HUNT Methodology



Source: Capcom

Methodologies



Right Click -> Send-To Methodology Section

Filter: Hiding out of scope items; hiding script, XML, CSS, general text, image, flash and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
22	https://www.appsecusa.org	GET	/	<input type="checkbox"/>	<input type="checkbox"/>	302	431	HTML		302 - Redirect
25	https://2017.appsecusa.org	GET	/	<input type="checkbox"/>	<input type="checkbox"/>	200	39193	HTML		Ap
59	https://2017.appse		https://2017.appsecusa.org/?id=test	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	39193	HTML		Ap

Request Response

Raw Headers Hex HT

HTTP/1.1 200 OK
Date: Fri, 22 Sep 2017
Server: Apache/2.4.1
Strict-Transport-Security: max-age=1000, path=/, secure
Set-Cookie: wfvt_224
Link: <https://2017.
Link: <https://2017.
Vary: Accept-Encoding
Content-Length: 3871
Connection: close
Content-Type: text/html; charset=UTF-8

Right-click context menu for item 59:

- Remove from scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder ⌘+^+I
- Send to Repeater ⌘+^+R
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser
- Send to HUNT – Methodology
- Engagement tools
- Show new history window
- Add comment
- Highlight
- Delete item
- Clear history
- Copy URL
- Copy as curl command
- Copy links
- Save item
- Proxy history help

Sub-menu for "Send to HUNT – Methodology":

- Account
 - Account Registration
 - File Download/Upload
 - Account Recovery
 - Money Transactions
 - Authentication
 - Search
 - Contact Us
 - General
 - API
- Insecure Direct Object Reference
- Cross Site Request Forgery
- Authentication Bypass – Vertical
- Cross Site Scripting
- SQL Injection
- Authentication Bypass – Horizontal

Description

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

HUNT – Methodology

Functionality

Account

- Insecure Direct Object Reference
- Cross Site Request Forgery
- Authentication Bypass – Vertical
- Cross Site Scripting
- SQL Injection**
- Authentication Bypass – Horizontal

- Account Registration
- File Download/Upload
- Account Recovery
- Money Transactions
- Authentication
- Search
- Contact Us
- General
- API

Settings

Description Bugs Resources Notes

Check all parameters that present themselves as an INSERT, DELETE, or UPDATE statement.

Multiple Request/Response

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

▼ HUNT – Methodology

 ▼ Functionality

 ▼ Account

 Insecure Direct Object Reference

 Cross Site Request Forgery

 Authentication Bypass – Vertical

 Cross Site Scripting

 SQL Injection

 Authentication Bypass – Horizontal

 ► Account Registration

 ► File Download/Upload

 ► Account Recovery

 ► Money Transactions

 ► Authentication

 ► Search

 ► Contact Us

 ► General

 ► API

 Settings

Description	Bugs	Resources	Notes
0 x	1 x	2 x	

Request Response

```
GET /?id=test HTTP/1.1
Host: 2017.appsecusa.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:55.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: wfvt_2244512973=59c47761a5546; _ga=GA1.2.1802552544.150604
_gid=GA1.2.1142792024.1506047738; wordfence_verifiedHuman=3778e224
_bizo_bzid=e2f668d7-9237-4d76-9ccf-30202c38ae2c; _bizo_cksm=90CDD1
_bizo_np_stats=155%3D1354%2C
Connection: close
Upgrade-Insecure-Requests: 1
```

Resources

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

HUNT – Methodology

- Functionality
 - Account
 - Insecure Direct Object Reference
 - Cross Site Request Forgery
 - Authentication Bypass – Vertical
 - Cross Site Scripting
 - SQL Injection**
 - Authentication Bypass – Horizontal
 - Account Registration
 - File Download/Upload
 - Account Recovery
 - Money Transactions
 - Authentication
 - Search
 - Contact Us
 - General
 - API
 - Settings

Description Bugs Resources Notes

<http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
<http://pentestmonkey.net/cheat-sheet/sql-injection/oracle-sql-injection-cheat-sheet>

Notes

Target	Proxy	Spider	Scanner	Intruder	Repeater	Sequencer	Decoder	Comparer	Extender									
<table border="1"><tr><td rowspan="2"><ul style="list-style-type: none">▼ HUNT – Methodology▼ Functionality ▼ Account<ul style="list-style-type: none"> Insecure Direct Object Reference Cross Site Request Forgery Authentication Bypass – Vertical Cross Site Scripting SQL Injection Authentication Bypass – Horizontal▶ Account Registration▶ File Download/Upload▶ Account Recovery▶ Money Transactions▶ Authentication▶ Search▶ Contact Us▶ General▶ API Settings</td><td>Description</td><td>Bugs</td><td>Resources</td><td>Notes</td></tr><tr><td colspan="4"><ul style="list-style-type: none">- Try and pop the SQLi on your own, you n00b- If all else fails, try SQLmap- Get help from Jason because he's an uber l33t h4x0r</td></tr></table>										<ul style="list-style-type: none">▼ HUNT – Methodology▼ Functionality ▼ Account<ul style="list-style-type: none"> Insecure Direct Object Reference Cross Site Request Forgery Authentication Bypass – Vertical Cross Site Scripting SQL Injection Authentication Bypass – Horizontal▶ Account Registration▶ File Download/Upload▶ Account Recovery▶ Money Transactions▶ Authentication▶ Search▶ Contact Us▶ General▶ API Settings	Description	Bugs	Resources	Notes	<ul style="list-style-type: none">- Try and pop the SQLi on your own, you n00b- If all else fails, try SQLmap- Get help from Jason because he's an uber l33t h4x0r			
<ul style="list-style-type: none">▼ HUNT – Methodology▼ Functionality ▼ Account<ul style="list-style-type: none"> Insecure Direct Object Reference Cross Site Request Forgery Authentication Bypass – Vertical Cross Site Scripting SQL Injection Authentication Bypass – Horizontal▶ Account Registration▶ File Download/Upload▶ Account Recovery▶ Money Transactions▶ Authentication▶ Search▶ Contact Us▶ General▶ API Settings	Description	Bugs	Resources	Notes														
	<ul style="list-style-type: none">- Try and pop the SQLi on your own, you n00b- If all else fails, try SQLmap- Get help from Jason because he's an uber l33t h4x0r																	

Save/Load JSON File

Target Proxy Spider Scanner Intruder Repeater Sequencer

- ▼ HUNT – Methodology
 - ▼ Functionality
 - ▼ Account
 - Insecure Direct Object Reference
 - Cross Site Request Forgery
 - Authentication Bypass – Vertical
 - Cross Site Scripting
 - SQL Injection
 - Authentication Bypass – Horizontal
 - Account Registration
 - File Download/Upload
 - Account Recovery
 - Money Transactions
 - Authentication
 - Search
 - Contact Us
 - General
 - API
 - Settings

Load JSON File

Save JSON File

Methodology Implementation

```
IExtensionStateListener, IContextMenuFactory, ITab

def createMenuItems(self, invocation):
    functionality = self.checklist["Functionality"]

        # Create the menu item for the Burp context menu
        bugcatcher_menu = JMenu("Send to HUNT - Methodology")

        for functionality_name in functionality:
            vulns = functionality[functionality_name]["vulns"]
            menu_vuln = JMenu(functionality_name)

            # Create a menu item and an action listener per vulnerability
            # class on each functionality
            for vuln_name in vulns:
                item_vuln = JMenuItem(vuln_name)
                menu_action_listener = MenuActionListener(self.view, self.callbacks, request_response,
functionality_name, vuln_name)
                item_vuln.addActionListener(menu_action_listener)
                menu_vuln.add(item_vuln)

            bugcatcher_menu.add(menu_vuln)

    burp_menu = []
    burp_menu.append(bugcatcher_menu)

    return burp_menu
```

Live Demo: Part Deux

DEMO GODS



**PLEASE MAKE THIS DEMO
WORK**

memegenerator.net

Plugin Installation



Source: [github](#)

Installation - Jython

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender

Extensions BApp Store APIs Options

 **Settings**

 This setting controls how Burp handles extensions on startup.

Automatically reload extensions on startup

 **Java Environment**

 These settings let you configure the environment for executing extensions that are written in Java. If loaded.

Folder for loading library JAR files (optional):

 [Select folder ...](#)

 **Python Environment**

 These settings let you configure the environment for executing extensions that are written in Python implemented in Java.

Location of Jython standalone JAR file:

 [Select file ...](#)

Folder for loading modules (optional):

 [Select folder ...](#)

Installation - Plugin

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts HUNT – Scanner

Extensions BApp Store APIs Options

Burp Extensions

Extensions let you customize Burp's behavior using your own or third-party code.

Add Loaded Type

Remove Java
Up Python
Down Python
Python
Python

Details Output Errors

Extension loaded

Name: HUNT – Scanner

Item

Extension type
Filename
Method
Extension state listeners
Suite tabs
Scanner checks

Please enter the details of the extension, and how you would like to handle standard output and error.

Load Burp Extension

Extension type: Python

Extension file (.py)

Please select a file

Look In: HUNT

conf license
images README.md
HUNT Slides.pdf
hunt.log
hunt_methodology.py
hunt_scanner.py

Standard Output

Output to sys
Save to file:
Show in UI

Standard Error

File Name:
Files of Type: All Files

Save Cancel

Cancel Next

Setting Target Scope

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User op

Site map Scope

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. All fields browse to your target and use the context menus in the site map to include or exclude URL paths.

Include in scope

Add	Enabled	Protocol	Host / IP range	Port	File
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Any	appsecusa		

Exclude from scope

Add	Enabled	Protocol	Host / IP range	Port	File
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Any			logout
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Any			logoff
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Any			exit
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Any			signout

Setting Passive Scanner Scope

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Issue activity Scan queue Live scanning Issue definitions Options

 **Live Active Scanning**

 Automatically scan the following targets as you browse. Active scan checks send various malicious requests designed to identify con

Don't scan
 Use suite scope [defined in Target tab]
 Use custom scope

 **Live Passive Scanning**

 Automatically scan the following targets as you browse. Passive scan checks analyze your existing traffic for evidence of vulnerabilities

Don't scan
 Scan everything
 Use suite scope [defined in Target tab]
 Use custom scope

Running the Passive Scanner

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options Use

Site map Scope

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding entire page content

Context menu for selected items:

- Add to scope
- Remove from scope
- Spider selected items
- Actively scan selected items
- Passively scan selected items**
- Send to HUNT – Methodology
- Engagement tools
- Compare site maps
- Expand branch
- Collapse branch
- Delete selected items
- Copy selected URLs
- Copy links in selected items
- Save selected items
- Issues
- View
- Show new site map window
- Site map help

	Method	URL	Param:
2015.appsec...	GET	/	
2016.appsec...	GET	/	
2017.appsec...	GET	/	
2014.appsec...	GET	/2014/	
2014.appsec...	GET	/2014/?wordfence_lo...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/p...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/p...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/p...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/t...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/t...	<input checked="" type="checkbox"/>
2014.appsec...	GET	/2014/wp-content/t...	<input checked="" type="checkbox"/>

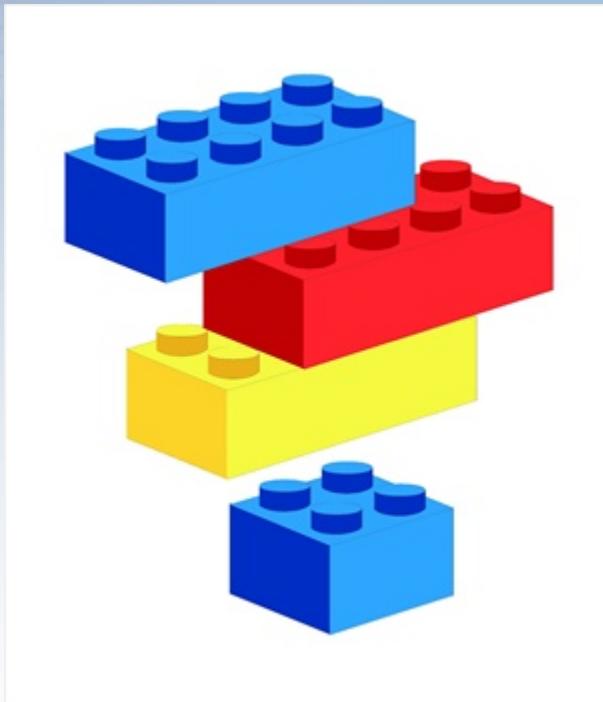
Response

Params Headers Hex

HTTP/1.1
2015.appsecusa.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:55.0) Gecko/20100101



Extensibility



Source: logic-canvas.com

Scanner Extensibility

Choose your own
issue type

CVE

Creating new issue
checks are as simple
as adding to the JSON
file.

```
{  
    "issues": {  
        "OS Command Injection": {  
            "check_location": {  
                "request": true,  
                "response": false  
            },  
            "detail": "HUNT located the <b>$param$</b> parameter  
inside of your application traffic. The <b>$param$</b>  
parameter is most often susceptible to OS Command Injection.  
HUNT recommends further manual analysis of the parameter in  
question.<br><br>For OS Command Injection HUNT recommends the  
following resources to aid in manual testing:",  
            "level": "Information",  
            "name": "Possible OS Command Injection",  
            "params": [  
                "daemon",  
                "upload",  
                "dir",  
                "execute",  
                "download",  
                "vulnerable_parameter"  
            ]  
        }  
    }  
}
```

Methodology Extensibility

Choose your own
ADVENTURE

Creating new methodologies are as simple as adding to the JSON file.

```
{  
  "checklist": {  
    "Settings": "",  
    "Functionality": {  
      "NEW METHODOLOGY SECTION": {  
        "description": "",  
        "tests": {  
          "Authentication Bypass - Vertical": {  
            "description": "Check to see if the login sequence  
can be bypassed in any way to get higher level permissions.",  
            "resources": [],  
            "bugs": [],  
            "notes": ""  
          }  
        }  
      }  
    }  
  }  
}
```

Live Demo: Electric Boogaloo

DEMO GODS



**PLEASE MAKE THIS DEMO
WORK**

memegenerator.net

The Future

- More built-in methodologies
 - OWASP, PCI, HIPAA, CREST, PTES
- Port to ZAP?
- More scanner checks/vulnerability classes
- More resources
- Dynamic JSON structure support
- Perfect GUI lol
- REST Support
- ~~Full Burp helpers (right click, search, highlight, etc)~~
- Resource/File name analysis (Instead of params)
- Alerts on content types (XML, JSON, Multipart-form)
- Response analysis alerts (errors)

Questions?

<https://www.github.com/bugcrowd/hunt>

