Pentesting With Burp Suite

Taking the web back from automated scanners





Outline

- Intro to Web App Testing
- Scoping with Burp
- Mapping with Burp Spider, Intruder, and Engagement Tools
- Replacing Some good common methodology tasks
- Automated Scanner Breakdown
- Stealing from other tools and Modifying your Attacks
- Fuzzing with Intruder and FuzzDB
- Auth Bruting with Burp Intruder
- Random Burping, IBurpExtender ++

Intro's

Jason Haddix

- Web App Pentester HP Application Security Center
 - GSEC, GPEN, GWAPT, blah, blah....
 - @jhaddix

Joel Parish

Web App Pentester – Redspin, Inc

Web App Pentests!

Process =

- Scoping -> Initial site recon, determine how large the application is, how dynamic, try to assess platform, etc. The age old question, engineer or sales guy?
- Pricing -> Use your scope to fit your assessment into a pricing model. Usually by days of analysis.
- Analysis/Hacking -> Get your hack on. Usually good to have a methodology.
- Reporting -> /sigh ... I mean, SUPER IMPORTANT, convey business risk, etc.

Burp Suite!

- Most commonly used interception proxy for web hackery. Pay tool with Free Version.
- Comprised of several parts:
 - Proxy Intercept and Log Requests
 - Spider Discover Content
 - Scanner App Vuln Scanner
 - Intruder Attack Tool
 - Repeater Attack Tool
 - Sequencer Token Assessment
 - Decoder & Comparer Auxiliary Tools



burp intruder repeater window help										
target proxy spider scanner intruder r	repeater sequencer	decoder comparer option:	alerts							
site man scone										
										_
Filter: hiding out of scope and not found items; hidin	ig CSS, image and genera	binary content; hiding 4xx respo	nses; hiding	empty fold	ers					
	host meth	od URL	parar	ms status	length	MIME typ	be title	comment	time requested	t
$\vdash \Box_{I}$	http://thepirateb GET	1		200	12653	HTML	Download music, movies, games,		14:01:01 21 Oct.	🔺
	http://thepirateb GET	/a.html		200	765	text			14:03:38 21 Oct.	=
	http://thepirateb GET	/about		200	11793	HTML	The Pirate Bay - The world's most		14:03:21 21 Oct.	
	http://thepirateb GET	/about.php		200	11788	HTML	The Pirate Bay - The world's most		14:03:28 21 Oct.	
	http://thepirateb GET	/ajax_details_filelist.php		200	496	HTML			14:26:47 21 Oct.	
ajax_details_artinfo.php	http://thepirateb GET	/ajax_post_comment.php		200	360	HTML			14:26:47 21 Oct	
jajax_details_filelist.php	http://thepirateb POST	/ajax_post_comment.php	×	200	360				14:05:58 21 Oct.	
🗣 🛞 ajax_post_comment.php	http://thepirateb GET	/blog		200	30854	HTML	The Pirate Bay - The world's most		14:03:21 21 Oct.	
- 🗋 blog	http://thepirateb GET	/blog/		200	30854	HTML	The Pirate Bay - The world's most		14:03:16 21 Oct.	
e 📄 blog	http://thepirateb POST	/blog/175	v	200	102613	HTML	The Pirate Bay - The world's most		14:06:01 21 Oct.	
- D browse	http://thepirateb GET	/blog/175		200	102613	HTML	The Pirate Bay - The world's most		14:05:32 21 Oct.	
		1								
	response request									
	Comment And And And									
e 💓 contact.php	raw headers he	x ntmi render								
🗠 🌺 details.php	HTTP/1.1 200 OK									-
- 🗋 doodles	X-Powered-By: PH	P/5.3.3								=
- D downloads	Expires: Mon, 26	Jul 1997 05:00:00 GMT								
- D help	Last-Modified: 7	hu, 21 Oct 2010 21:00:	27 GMT							
	Cache-Control: n	o-store, no-cache, mus	-revalid	ate						
	Cache-Control: p	ost-check=0, pre-check	=0							
I language	Pragma: no-cache	(here here here a state HTTE O								
- 📋 legal	Vorue Accept Free	xt/ntml;charset=01F-8								
🗣 🌺 login	Date: Thy 21 Oc	- 2010 21.00.27 CWT								
🗣 🧰 music	Server: lighttpd	C 2010 21:00:27 GMI								
- 🗋 music	Content-Length:	12271								
	concent hengen.	15571								
- Cont	DOCTYPE html F</td <td>UBLIC "-//W3C//DTD XHT</td> <td>ML 1.0 Tr</td> <td>ansitio</td> <td>nal//EN"</td> <td></td> <td></td> <td></td> <td></td> <td></td>	UBLIC "-//W3C//DTD XHT	ML 1.0 Tr	ansitio	nal//EN"					
	"http://www.w3.c	rg/TR/xhtml1/DTD/xhtml	l-transit	ional.dt	:d">					
	<pre></pre>									
e 🞯 recover	<head></head>									
register	<pre><title>Download</title></pre>	music, movies, games,	software!	The Pir	ate Bay	– The	world's most resilient BitT	orrent site </td <td>title></td> <td></td>	title>	
S robots.ht										
rss	title="Search The Pirate Bay" />									
🕨 🔶 🧰 s	<pre><meta http-equiv<="" pre=""/></pre>	="Content-Type" conten	="text/h	tml; cha	arset=ut	f-8"/>				-
🗣 🧰 search	+ < >								0 mate	hes

Utilizing Burp in Process!

Lets start with the Process:

- Scoping: Defining the range of the test. Leads to pricing.
- Spidering gives us a site map. We want to determine application complexity by how much dynamic content there is.

- http://thepirateba	1	•	host me	
$-\Gamma_{1}$	http://thepiratebay.org/			http://thepirateb GET
- D a.html	add item to scope	1		http://thepirateb GET
- D about	romovo itom from ocono			http://thepirateb GET
	Ternove itern from scope			http://thepirateb GET
	spider this host			http://thepirateb POS
ajax_deta	actively scan this host			http://thepirateb GET
► 🦉 ajax_pos	penaivaly acception beat			http://thepirateb GET
- Dlog	passively scan this host			http://thepirateb PO
🗠 🚞 blog	engagement tools 💦 🕨 🕨	S	ear	ch
🗠 🧰 browse	expand branch	fi	nd	comments
- D browse	expand requested items	fi	nd	scripts
- Contact	collanse branch	fi	nd	references
🗠 🦉 contact.p			nal	ve a torget
details.pl	delete host	d	па	ysetarget
- 🗋 doodles	copy URLs in this host	d	isc	over content
- 🗋 download	copy links in this host	s	che	dule task
help	save selected items	s	im	late manual testing

Utilizing Burp in Process!

Right click on your domain -> Engagement tools -> Analyze Target & Find Scripts. (Spider 1st).

ē 23

(3) target analyser | http://thepiratebay.org/ dvnamic URLs

Number of dynamic URLs:	24
Number of static URLs:	67806
Total number of parameters:	40
Number of unique parameter names:	20

static URLs

Note: This analysis is based on the current contents of the site map, and no new requests have been made. Only parameters within the query string and request body are included in the analysis. URLs identified as "static" are those which do not take any parameters, though their responses may still be dynamically generated.

save report

summarv

This gives us a better idea (sometimes only pre-authentication) how to budget/price the assessment. Spidering is not illegal. Throttle if necessary. So easy even a sales guy can do it!

Utilizing Burp* in Analysis!

Analysis = Hackery

- Usually follows a "methodology":
 - Open Source Intelligence Gathering
 - Mapping the target *
 - Vulnerability Assessment & Fuzzing *
 - Exploitation *
 - Session Testing *
 - Authentication Testing *
 - Logic Testing
 - Server Tests *
 - Auxiliary tests (Flash, Java, ActiveX, Web Services)
 - + more... many people do different things or do their tests in different orders. *

Burp Intruder Payload Types

- Sniper sends a single payload to each of the selected parameters; i.e. each parameter is sequentially tested with the same set of variables
- Battering ram sends a single payload to all of the selected parameters at once; i.e. all parameters will be passed the first variable, followed by all Parameters being passed the second variable, and so on until the payload is completed.
- Pitchfork sends a specific payload to each of the selected parameters; i.e. all parameters need to be passed its own payload, and the variables of each payload are passed to its designated parameter in sequence.
- Cluster bomb starts with a specific payload to each parameter, and when all variables have been tested, will start testing with the payload from the next variable, such that all parameters get tested with all variables
- For big lists use "runtime file" Payload set...

Burp Mapping!

- Burp Spider will discover all readily available linked content. Make sure you walk the app as well.
- We also want to indentify hidden or non-linked content, normally using tools like:
 - Dirbuster (OWASP)
 - Wfuzz (Edge Security)
- Burp Suite has its own functionality for this!
 - Right click on your domain ->
 Engagement tools -> Discover Content

http://thepirate	ebay.org	-	host			
- []/	http://thepiratebay.org/		http://thepirateb			
- 🗋 a.htm	intpirtitopiratosa, io. g.		http://thepirateb			
	add item to scope		http://thepirateb			
	remove item from scope		http://thepirateb			
	anidar this hast		http://thepirateb			
	spider this host		http://thepirateb			
- 🗋 ajax_	actively scan this host		http://thepirateb			
🗠 🎯 ajax_	passively scan this host		http://thepirateb			
- D blog						
	engagement tools	search				
brow:	expand branch	find comments				
- 🗋 brow:	expand requested items	find scripts				
- 🗋 conta	collapse branch	find references				
🗠 🎯 conta	delete host	analyse target				
🗠 🛞 detail	copy URLs in this host	discover content				
- 🗋 dood	convilinke in this bast	schedule task				
— 🗋 down	copy mines in this host	Schedule task				
- Cì help	save selected items	simulate manual testing				

Burp Mapping!

We can also steal Dirbuster's and Wfuzz's directory lists and use them with Burp Intruder for better coverage if needed.

Dirbuster has the best lists:

📄 big_dirb.txt	6/3/2004 8:38 PM	TXT File	22 KB
directory-list-2.3-big.txt	8/14/2008 3:05 PM	TXT File	14,456 KB
directory-list-2.3-medium.txt	8/14/2008 3:05 PM	TXT File	1,934 KB
extensions_common.txt	10/19/2003 1:56 PM	TXT File	1 KB
file_enumeration.conf	7/18/2008 12:05 PM	CONF File	9 KB

Set up an intruder attack like so...

Burp Mapping!

Intruder - Burp can use Dirbuster/Wfuzz lists.

- Right Click "/" and "Send to Intruder"
- In the "Positions" tab Use Sniper Payload
- Put the \$\$'s after "/"
- Under "Payloads" tab
 - Use "Preset List" → Click "load" Choose
 - a Dirbuster List or wfuzz list.

*** Quick tip, shutout the noise from other sites your browser is interacting with by setting up a scope for the proxy tab: Right Click your domain -> "add item to scope" -> Right click on the filter bar -> show only in scope items... that's better! ***



Burp Mapping++ !

Other mapping activities?

- Look for administrative portals
 - We used to use a modified script: admin-scan.py
 - Easily ported to burp intruder using the method on the last slide
 - <u>http://xrayoptics.by.ru/database/</u>
 - Tons of little scanners and useful tools here...

admin_path = ['manager/','manager/html/upload','web-console/ServerInfo.jsp','jmx-console/',
'CFIDE/administrator/enter.cfm','CFIDE/componentutils/login.cfm','admin.php','admin/','administrator/','moderator/',
'webadmin/','adminarea/','bb-admin/','adminLogin/','admin_area/','panel-administracion/','instadmin/','memberadmin/',
'administratorlogin/','adm/','admin/account.php','admin/index.php','admin/login.php','admin/admin.php',

Although not in this phase of the assessment server content and vuln/server checks (a la Nikto) can be done this way!

Now we move on...





Scanners!

Scanners!

- Save time and money.
- Good first step in application security.
- Have lots of vetted code, attack strings, detection regex's, auxiliary tools, teams to support and update etc...



Commercial:

Open-Source:

- Acunetix
- Appscan
- WebInspect
- Netsparker
- Burp Scanner
- Nessus
- CORE
- Cenzic
- many more...

- w3af
- Wapiti
- Grendel Scan
- Nikto
- Websecurify
- Skipfish
- Metasploit Wmap
- Wfuzz
- CAT
- many more...

Scanners!

Scanners

- Lots of application assessment is based around fuzzing application input points.
- Bruteforce fuzzing vs intelligent fuzzing
 - Identify input points
 - Does this functionality display something back to the user?
 - Does it interact with a database?
 - Does it call on the server file system?
 - Does it call on a URL or external/internal site/domain?
 - Inject large amounts of arbitrary data (fuzzing) or inject large amounts of relevant attacks strings (intelligent fuzzing)

Predominantly this is what most scanners do... The kitchen sink approach.

Be a ninja... not a monkey

If you're a pentester... don't be this:



Burp VA and Scanning!

- 1st off Burp has it's own scanner, so... win. (it's pretty good)
- If web app scanners just use a grip of attack strings on known input points, why cant we do this manually with Burp Intruder?
 - We most certainly can!

Enter... the fuzzdb by



"Categorized by platform, language, and attack type, enumeration and attack patterns have been collected into highly injectable fuzz payload lists. fuzzdb contains comprehensive lists of <u>attack payloads</u> known to cause issues like OS command injection, directory listings, directory traversals, source exposure, file upload bypass, authentication bypass, http header crlf injections, and more. Since system responses also contain predictable strings, fuzzdb contains a <u>set of regex pattern dictionaries</u> such as interesting error messages to aid detection software security defects, lists of common Session ID cookie names, and more."

Fuzzdb!

Think of it as a set of ultimate web fu cheatsheets...

	discovery	# credit to rsnake	'sqlvuln
	uiscovery	<script>alert('XSS');</script>	'+sqlvuln
	file upload	'';!" <xss>=&{()}</xss>	sqlvuln;
n.	former at station and	<script src="http://ha.ckers.org/xss.js"></script>	(sqlvuln)
	format strings		a' or 1=1
	http protocol		"a"" or 1=1"
	internet and flam.		a = a
	Integer overriow		1 or 1=1
	Idap		a' waitfor delay '0:0:10'
		<img src="javascript:alert(String.fromCharCode(88,83,83)</th"/> <th>1 waitfor delay '0:0:10'</th>	1 waitfor delay '0:0:10'
	misc - payloads	SRC= <img< th=""><th>declare @q nvarchar (4000) select @q =</th></img<>	declare @q nvarchar (4000) select @q =
	misc - wordlists	6;avascript&	0x770061006900740066006F0072002000640065006C00610079002
n.	an annual succession	1;>	0
	os command execution	<img< th=""><th>031003000270000</th></img<>	031003000270000
	os directory indexing	SRC=javas�	<pre>declare @s Varchar(22) Select @s = 0v77616074666677220646566617020272020202020202020200</pre>
n.	and have a second	8ert('X	0x776169746668722064656061792027505A505A51502700 exec(6 0x730065006c0065006300740020004000400076006500720073006
	path traversal	<img< th=""><th>declare @s varchar (8000) select @s = 0x73656c656374204</th></img<>	declare @s varchar (8000) select @s = 0x73656c656374204
	rfi	SRC=javascript	exec(@s)
n.	and the instruction		a'
	server side includes		?
	source disclosure		' or 1=1
	cal injection	<img src="jav	
ascript:alert('XSS');"/>	or 1=1		
	sqi injection		x' AND userid IS NULL;
	usernames and passwords	<script src="http://ha.ckers.org/xss.js" xss=""></script>	X' AND email IS NULL;
	-	<script src="http://ha.ckers.org/xss.js?<B"></script>	

Fuzzdb!

- The fuzzdb gives us a good starting point... why not parse and add all those open source scanner attack strings too? (fuzzdb has done 'some' of this)
- Most of them are plaintext resource files that the scanners call on... easy to parse and add to our modified fuzzdb.
- <.< >.> Shifty eyes...
- Keeping attacks separate via vector (SQLi, XSS, LFI/RFI, etc...) allows us to make less requests because as humans we know what type of attack we are looking to achieve and we can limit Burp to that subset of attacks.
- Our set of attack strings + burp files will be released a few days post con, or put directly into the fuzzdb trunk (whichever happens 1st;)

Interpreting fuzz results

- Usually when fuzzing we can use response size, return time, and regex's to look for fishy application behavior.
- Fuzzdb has a great Burp grep file:
 - Open Burp Suite, go to the Intruder tab, and the Options sub-tab
 - Look for the section "grep"
 - Click "clear" to clear the existing listings in the list box
 - Click "load" and load regex/errors.txt from your fuzzdb path, as below
 - This will search all output pages generated by Intruder payloads for the extensive list of known error strings, for later analysis.
- After successful identification, using Burp or auxiliary tools/scripts for exploitation is easy...
- Filter Evasion? Old blacklists never learn new tricks =(
 - http://www.wiretrip.net/rfp/txt/whiskerids.html
 - http://www.securityaegis.com/filter-evasion-houdini-on-the-wire/

Burp Session Testing

- Usually session tokens from common frameworks are well vetted but in instances where you see a custom session token fly by Burp's Sequencer can gather and test for entropy via all kinds of compliance needs.
- Pretty reporting graphs.

GET request to http://news.bbc.co.uk/
send to intruder
send to repeater
send to spider
send to sequencer
send request to comparer
send response to comparer
copy URL to clipboard
copy all URLs to clipboard

Overall result

The overall quality of randomness within the sample is estimated to be: excellent. At a significance level of 1%, the amount of effective entropy is estimated to be: 146 bits

Effective entropy

The chart shows the number of bits of effective entropy at each significance level, based on all tests. Each significance level defines a minimum probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occuring falls below this level, the hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to relet the hypothesis that the sample is random, and so increases the chance that non-random data will be treated as random.



Burp Auth Testing

Bruteforcing Authentication with Burp Intruder

- Attempt Login
- Go to Proxy History Tab
- Find the POST request
- Send to Intruder
- Use Cluster Bomb payload
- Clear all payload positions
- Mark username and password fields as payload positions
- Goto "payloads" tab
- Set "payload set" 1 to your username list
- Set "payload set" 2 to your password list
- Click on the intruder Menu
- Start Attack
- Look for different lengths or grep possible successful auth messages under options

394 http://thepiratebay.org

POST /login

attack type cluster bomb

2 payload positions POST /login HTTP/1.1 Host: thepiratebay.org User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;o Accept-Language: en-us,en;q=0.5 Accept-Encoding: gzip,deflate Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7 Keep-Alive: 115 Proxy-Connection: keep-alive Referer: http://thepiratebay.org/login Cookie: PHPSESSID=434fc7158c355611491966f7f6alc475; language=en_EN Content-Type: application/x-www-form-urlencoded Content-Length: 52

username=ScheckS&password=ScheckS&act=login&submit=Login

Burp Auth Testing

The password lists are non extensive!

- Go thank Ron, he makes Facebook cry:
 - http://www.skullsecurity.org/blog/2010/the-ultimate-faceoffbetween-password-lists
 - Huge password repository. Actual user data from hacked sites:
 - RockYou
 - Phpbb
 - Myspace
 - Hotmail
 - Hak5
 - Facebook
 - More...
- @iagox86

stupididiot strong997 stratos65 stormsuper191091 stephen stelios1974@ stefanzvanialehandrepato steaua start1 star2000 star123 st667m st117h ssurup 333333 3333 ssosdaasjd ssetiram123 srs090480 sriramcentre sreeharipovvil srbijakula srbija1 srbija sr2603 sqyatch22 spirit10* spenc3r1*

Random Burping Tips

- Burp Spider in conjunction with - Engagement Tools → Search makes Burp an IH tool
- Find injected code or javascript redirects when inspecting a compromised site/app.

🛞 burp sui	ite search http://thep	iratebay.org/								x
search			options	in-scope only	locations	₽ res	sponse hea	tools		
		go	regex	dynamic update	request body	✓ res	sponse bod	y roxy	✓ repeater	
source target target target target target target target target target	host http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or http://thepiratebay.or	rg , rg , rg , rg , rg , rg , rg , rg ,	/ /about /blog /browse /contact /doodles /downloads /help /language /language	URL		status 200 200 200 200 200 200 200 200 200 20	length 13096 11851 23382 12666 12130 20397 13566 14656 14656 17119 15357	time requested 11:39:56 9 Aug 11:48:58 9 Aug 11:50:26 9 Aug 11:50:27 9 Aug 11:48:58 9 Aug 11:48:58 9 Aug 11:48:57 9 Aug 11:48:57 9 Aug 11:48:58 9 Aug		
target http://thepiratebay.org /login 200 9847 11:48:58 9 Aug response request (a) htef="http://slopsbox.com" title="SlopsBox" target="_blank">SlopsBox (a) htef="http://slopsbox.com" title="SlopsBox" target="_blank">SlopsBox (a) htef="http://baywords.com" title="BayWords" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (a) htef="http://baywords.com" title="PateBay" target="_blank">BayWords (b) (a) htef="http://www.pirateshops.com" title="PateBay" target="_blank">ImacWords (b) (a) (b) (b)<										
Pirate Chat, a, b, b,										

Random Burping Tips

- Proxy Tab --> Options
- Disable clientside input validation when testing via the browser.
- Unhide hidden form fields.



IBurpExtender

- Hooks into HTTP Request for pre/post Burp processing
- Edit Burp configuration pragmatically
- Send requests to repeater/intruder
- Access to scanning/proxy data

Eww Java

Do I have to work with Java? -Xmn4096M -Xms4096M -Xmx4096M Java is fast now And the JVM is awesome

	compare 2	-		25%	median
	C GNU gcc	1.00	1.00	1.00	1.10
${\color{black}\overline{\checkmark}}$	C++ GNU g++	1.00	1.00	1.00	1.11
	ATS	1.00	1.00	1.00	1.12
	Ada 2005 GNAT	1.00	1.00	1.17	1.98
${\color{black}\overline{\checkmark}}$	Java <mark>6 steady state</mark>	1.10	1.10	1.12	2.00
	Haskell GHC	1.23	1.23	1.48	2.13
	Go 6g 8g	1.30	1.30	1.56	2.32
	Scala	1.14	1.14	1.32	2.40
${\color{black}\overline{\checkmark}}$	Java 6 -server	1.11	1.11	1.63	2.51

JVM

 Lets you leverage agile synergies to arbitrate technical debt across organiznational and personal boundaries.
 Yuk

JVM

- Ruby (JRuby)
- Python (Jython)
- Javascript (Rhino)
- Clojure
- Scala

 And Lua, PHP (Quercus), COBOL ________ of and dozens of other languages.

Burp Extensions in other Languages

- <u>http://github.com/emonti/buby</u> (JRuby)
- http://blog.ombrepixel.com/public/burppytho n_v0.1.zip (Jython)
- Write your own! (all of the above JVM languages can use the IBurpExtender interface)

Things humans aren't good at

- I'm not a bit twiddling God
- GDS has done some great stuff with decompressing DEFLATE and binary SOAP HTTP requests/responses.
- Using JRuby/Buby to attack Java Object Serialization <u>https://media.blackhat.com/bh-eu-</u> <u>10/whitepapers/Saindane/BlackHat-EU-2010-Attacking-</u> <u>JAVA-Serialized-Communication-wp.pdf</u>

Things humans aren't good at

- Padding Oracle vulnerabilities
- Write a Burp hook to decrypt ASP.net viewstate with the machine key from the extracted from padding oracles.
 - Re-encrypt on exit
- Use Burp's built-in viewstate editor, edit flags and win!

Induces repeater window neipp arget proxy Spider scanner intruder repeater Geoder comparer comms alerts arget options history spider scanner intruder repeater Geoder comparer comms alerts sponse from http://www.nrc-recycle.org/80/Secure/Login.aspx [208.106.226.177] forward drop intercept is on action aw neaders hex html render viewstate intruder repeater intruder	Surp suite p burp intruder	rofessional v1.	2 vindout	oln	_								X
rget provy spider scanner intruder repeater sequencer decoder comparer comms alerts nercept options history sponse from http://www.nrc-recycle.org.80/Secure/Login.aspx [208.106.226.177] forward drop intercept is on action aw headers hex html render viewstate	burp intruder repeater window help												
Intercept Options history sponse from http://www.nrc-recycle.org.80/Secure/Login.aspx [208.106.226.177] forward drop intercept is on action aw headers hex html render viewstate List int 3 Pair mull List int 1 Pair tint 1 Stimg SiteMapProvider string mojosite1 Visit int 1 Pair Visit int 1 Pair Stimg SiteMapProvider string mojosite1 Visit SiteMapProvider string nojosite1 Visit Visit SiteMapProvider SiteMapProv SiteMap	target prox	y spider	scanner	intruder	repeater	sequencer	decoder	comparer	comms	alerts			
sponse from http://www.nrc-recycle.org.80/Secure/Login.aspx (208.106.226.177) forward drop intercept is on action aw headers hex html render viewstate	intercept	options hist	tory										
forward drop intercept is on action aw headers hex ntml render viewstate null List int 3 Pair Intercept is on <td>response fron</td> <td>n http://www.nr</td> <td>c-recycle.o</td> <td>rg:80/Secure</td> <td>e/Login.aspx</td> <td>(208.106.226</td> <td>.177]</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	response fron	n http://www.nr	c-recycle.o	rg:80/Secure	e/Login.aspx	(208.106.226	.177]						
aw headers hex html render viewstate List int 3 Pair null List int 1 Pair int 1 Pair int 1 Pair int 1 Pair int 1 int 1	forward	drop		intercept is	on	action							
Image: string bit string bit string bit string mojosite1 Image: string bit string bit string bit string mojosite1 Image: string mojosit string mojosite1 Image: string mojosit string mojosite1 Image: string mojosit string m	raw head	ers hex	html rei	nder viev	vstate								
Image: string stremap: rowide results Image: string rowide results		– nu	.11										
int 3 Pair null List • Pair • List • List • List • String BiteMapProvider • string mojosite1 • If 01 0f 0f 05 09 37 38 33 31 36 30 37 33 33 0f ŷ 783160733 64 16 02 66 0f 64 16 02 02 03 0f 64 16 08 02 01 d f d d 0f 16 04 1e 0f 53 89 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 67 4e 6f StartingNo		e- Lis	st										
			int 3										
Image: constraint of the string site MapProvider Image: constraint of the string site site site site site site site site		9	Pair										
			– null										
int 1 - Pair - List - string SiteMapProvider - string mojosite1 - string mojosite1			∳– List										
Pair List string SiteMapProvider string mojosite1 ff 01 0f 05 09 37 38 33 31 36 30 37 33 33 0f ŷ 783160733 64 16 02 02 03 0f 64 16 08 02 01 d d d 0f 16 04 1e 0f 53 69 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 67 4e 6f StartingNo			— in	t 1									=
Image: bit			φ− Pa	air									
string SiteMapProvider string mojosite1			9	- List									
string mojosite1 ff 01 0f 0f 05 09 37 38 33 31 36 30 37 33 33 0f ÿ 783160733 64 16 02 66 0f 64 16 02 02 03 0f 64 16 08 02 01 d f d d 0f 16 04 1e 0f 53 69 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 67 4e 6f StartingNo				- string	SiteMapPro	ovider							
ff 01 0f 05 09 37 38 33 31 36 30 37 33 33 0f ÿ 783160733 64 16 02 66 0f 64 16 02 02 03 0f 64 16 08 02 01 d f d d 0f 16 04 1e 0f 53 69 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 6f StartingNo				- string	mojosite1								
ff 01 0f 05 09 37 38 33 31 36 30 37 33 33 0f ÿ 783160733 64 16 02 66 0f 64 16 02 02 03 0f 64 16 08 02 01 d f d 0f 16 04 1e 0f 53 69 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 67 4e 6f StartingNo	•												
II U1 U1 U3 U3 U3 U3 U3 U1 U1 U3 U3<	0 4	04	of I		27	0 22		20 07		22		460722	
Of 16 04 1e Of 53 69 74 65 4d 61 70 50 72 6f 76 SiteMapProv 69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 6f StartingNo	1 64	16 02	66 0	05 09 0f 64	16 0	o 33 3 2 02 0	01 30 01 0f	64 16	08	02 01	y 783 d fd	d	
69 64 65 72 05 09 6d 6f 6a 6f 73 69 74 65 31 1e ider mojosite1 0f 53 74 61 72 74 69 6e 67 4e 6f StartingNo	2 Of	16 04	1e (Of 53	69 7	4 65 4	4d 61	70 50	72	6f 76	Site	MapProv	
Of 53 74 61 72 74 69 6e 67 4e 6f StartingNo	3 69	64 65	72 0	05 09	6d 61	f 6a 6	of 73	69 74	65	31 1e	ider moj	osite1	
	4 Of	53 74	61 7	72 74	69 6	e 67 4	le 6f				Starting	No	
		1 - 100											

Turning Burp into an Automated Scanner?

- Paul Haas's sodapop tool uses Burp Headless to spider a website and actively scan for vulnerabilities, and to log everything to stdout.
- (http://www.redspin.com/blog/2010/09/20/advanced-burp-suiteautomation-2/)
- Easy to integrate into large collections of startup scans

\$./sodapop.sh www.example.com example "CookieMonster=LikesCookies" suite: method BurpExtender.processProxyMessage() found suite: method BurpExtender.processHttpMessageMethod() found suite: method BurpExtender.registerExtenderCallbacks() found suite: method BurpExtender.setCommandLineArgs() found suite: method BurpExtender.applicationClosing() found suite: method BurpExtender.newScanIssue() found proxy: proxy service started on port 8080 scanner: live active scanning is enabled - any in-scope requests made via Burp P suite: Attempting to restore state from 'sodacan.zip' proxy: proxy service stopped on port 8080 proxy: proxy service started on port 8080 scanner: live active scanning is enabled - any in-scope requests made via Burp P suite: Adding www.example.com to scope, spider and scanner suite: Including 'Cookie: CookieMonster=LikesCookies' to all in-scope requests. suite: Starting spider on http://www.example.com:80/ at Mon Sep 20 9:00:01 PDT 2 suite: Monitor thread started at Mon Sep 20 9:00:05 PDT 2010 and waiting for spi suite: Monitor thread started and waiting for spider to complete scanner: Low Cookie without HttpOnly flag set: http:/www.example.com:80/ scanner: High Cleartext submission of password: http://www.example.com:80/login/ scanner: Low Password field with autocomplete enabled: http://www.example.com:80 scanner: High XPath injection: http://www.example.com:80/api/access/

Turning Burp into an Automated Scanner?

- W3af, awesome Python web attack framework
- So, now we have access to Burp scanners/proxy, and a Python runtime. Why don't we just import w3af checks into burp? (http://blog.ombrepixel.com/post/2010/09/09/Run ning-w3af-plugins-in-Burp-Suite)

C:\Burp>java -Xmx512m -classpath burpsuite v1.3.03.jar;burppython.ja								
init: Boo	init: Bootstrapping class not in Py.BOOTSTRAP TYPES[class=class org.;							
BurpExter	nder.py needs to be in a folder	listed below:						
['C:\\Bur	cp\\Lib', '/C:/Burp/burppython.j	ar/Lib', ' classpath ', '						
loading w	Jaf plugins	·						
Loading g	grep.domXss	Success						
Loading g	grep.error500	Success						
Loading g	grep.errorPages	Success						
Loading g	grep.feeds	Success						
Loading g	grep.fileUpload	Success						
Loading g	grep.hashFind	Success						
Loading g	grep.httpAuthDetect	Success						
Loading g	grep.privateIP	Success						
Loading g	grep.ssn	Success						
Loading g	grep.strangeHeaders	Success						
Loading g	grep.strangeHTTPCode	Success						
Loading g	grep.strangeReason	Success						
Loading g	grep.svnUsers	Success						
Loading g	grep.wsdlGreper	Success						

Conclusions

- Be your own scanner
- Don't be a tool, really use your tools.
- Humans > machines



Closing Notes or Whatevs

Taking your mom back from automated scanners



