Overview!

Security work often requires system knowledge. In this lab, we will create a cheat sheet we can use when working on future labs and homework assignments. Please do your best to answer as many as possible on your own, but note that you can also use the synchronous Q&A session for help.

There are no restrictions on resources that can be used to answer questions. This studio will be graded for completeness (not correctness), and group submissions are allowed.

The windows column is optional. Windows users are encouraged to complete it.

#	**How do I?**	**… in Linux**	**… in Windows**
1	List directory contents?	ls -al	
2	Find my machine name?	uname -a	
3	Start an admin console session?	su	
4	Find which processes use the most CPU or memory?	top	
5	Stop/Kill a process?	kill <process ID>	
6	Find out how much disk space is free?	df -h	
7	Find out who is logged in?	w	
8	Find a log of recent logins and login attempts?	vi /var/log/auth.log	

9	Find my IP and MAC addresses?	ifconfig	
10	Examine my OS name and version?	lsb_release -a or hostnamectl	
11	Find kernel version?	hostnamectl	
12	Examine which programs run at system boot time?	ls -l /etc/init.d or sudo systemctl list-unit-files --type=service --state=enabled --all	
13	Stop a program from running at system boot time?	sudo systemctl disable servicename	
14	Find the list of trusted certificates installed on my system?	ls /etc/ssl/certs	
15	Remove a trusted certificate from my system?	$ rm /usr/local/share/ca-certificates/{unnecessary_certificate}.crt $ update-ca-certificates -fresh	
16	Compile my program?	c compiler: gcc test.cpp java compiler: javac test.java	
17	Display an object file?	objdump -d test.o	
18	Start gdb?	gdb <program>	N/A
19	gdb: Set a breakpoint?	b <function name>	N/A
20	gdb: Show registers information?	info reg	N/A
21	gdb: Present stack values?	info frame	N/A
22	gdb: Read stack content (explain in words)	`(gdb) info frame 4` `Stack frame at 0x7fffffffd9a0:`	

		rip = 0x400637 in main (example2.c:17); saved rip = 0x7ffff7a2d830 caller of frame at 0x7fffffffd970 source language c. Arglist at 0x7fffffffd990, args: argc=3, argv=0x7fffffffda78 Locals at 0x7fffffffd990, Previous frame's sp is 0x7fffffffd9a0 Saved registers: rbp at 0x7fffffffd990, rip at 0x7fffffffd998 1. Stack frame at 0x7fffffffd9a0: It means that this stack starts at the address 0x7fffffffd9a0. 2. rip = 0x400637 in main (example2.c:17): It means that 0x400637 is the next instruction address. 3. caller of frame at 0x7fffffffd970 : Indicates the frame that called the current frame 4. Arglist at 0x7fffffffd990 : Argument lists start at the address 0x7fffffffd990 5. Locals at 0x7fffffffd990 : Local variables start at the address 0x7fffffffd990	
23	Examine the program structure without the source file (explain in words)	Use a de-compiler for a executable file	
24	List all open network connections?	ss	
25	Find the process responsible for each open network connection?	ss -p ex) ss -p | grep firefox	

26	Find the binary executable responsible for each open network connection?	1. find the PID using ‘ss -p | grep firefox’ 2. It shows PID of firefox 3. And then ‘ps -aux <PID>’ to find out what the executable file is	
27	Reset my network interface?	sudo /etc/init.d/networking restart	
28	Find my default IP gateway?	ip route	
29	Find my default name server?	grep "nameserver" /etc/resolv.conf	
30	Examine contents of the ARP cache?	arp -a	
31	Add an entry to the ARP cache?	sudo arp -s <host ip> <mac address> ex) sudo arp -s 10.0.0.2 00:0c:29:c0:94:bf	
32	Examine contents of the DNS cache?	sudo killall -USR1 systemd-resolved sudo journalctl -u systemd-resolved > ~/dns-cache.txt less ~/dns-cache.txt	
33	Make a local DNS query respond with an IP of my choosing?	nslookup somewhere.com some.dns.server ex) nslookup google.com som.dns.server	
34	My favorite command-line editor?	vi	
35	Bring the most recent suspended job to the foreground?	fg	

36	List and resume stopped jobs in the background?	list : jobs resume : fg <job number>	
37	List files opened by processes?	lsof	