**The Big Five Extended**

**Processes Directories Files Users Logs | Services Modules Libraries Evidence Tools**

**Haste makes waste:** echo “Don’t Panic.”

**Processes**

**Large amounts of CPU/RAM:** top **Process tree:** ps –auxf **Open network ports or raw sockets:** netstat –nalp  
 netstat –plant  
 netstat -plunt  
 ss –a –e –i –n  
 lsof [many options]  
 lsof –p <pid> **Deleted binaries still running:** ls –alR /proc/\*/exe 2>/dev/null | grep deleted **Process command line/cmdline:** strings /proc/<pid>/comm  
 strings /proc/<pid>/cmdline **Real process path:** ls –la /proc/<pid>/exe **Process environment:** strings /proc/<pid>/environ **Process working directory:** ls –laR /proc/\*/cwd **Processes running from tmp, dev dirs:** ls –laR /proc/\*/cwd 2>/dev/null | grep tmp  
 ls –laR /proc/\*/cwd 2>/dev/null | grep dev

**Directories**

**Commonly targeted directories:** /tmp, /var/tmp, /dev/shm, /var/run,  
 /var/spool, /home/<user>, /root **List and delimit spaces, etc in names:** ls –lap **List all hidden directories:** find / -type d –name ‘.\*’ 2>/dev/null

**Files**

**Immutable files and directories:** lsattr –R / 2>/dev/null | grep ‘\----i’ **Find SUID/SGID files:** find / -type f -perm -4000 –perm -2000 –exec ls -lg {} \; **Files/dirs with no user/group name:** find /home/gel0 -nouser -nogroup -exec ls -lg {} \; **List all file types in current dir:** file \* -p

**Find executables:** find / -type f –exec file –p ‘{}’ \; | grep ELF **Find files with contents modified within last day:** find / -mtime -1 **Find files with changed attributes within last day:** find / -ctime -1 **Find hidden files:** find / -name ‘.\*’ –print **Package commands to find changed files:** rpm –Va | grep ^..5. (debsums –c)

**Users**

**Find all ssh authorized\_keys files:** find / -name authorized\_keys **History files for users:** find / -name ‘.\*history’ **History files linked to /dev/null:** ls –alR / 2>/dev/null | grep ‘\*.history’ | grep null **Look for UID 0/GID 0:** grep ‘:0:’ /etc/passwd **Check sudoers file (wheel = default admin group):** cat /etc/sudoers *and* /etc/group **Check scheduled tasks:** crontab [-u user] –l  
 for user in $(cut -f1 -d: /etc/passwd); do echo $user; crontab -u $user -l; done  
 atq  
 systemctl list-timers –all

**Logs**

**Check for zero size logs:** ls –al /var/log/\* **Dump login attempts:** utmpdump /var/log/wtmp *or* btmp  
 utmpdump /var/run/utmp **View login attempts:** last *or* last –n 50 *or* last <user>  
pam\_tally2 –u <user> **Logs by userid (if auditd is present):** ausearch –i –ui <userid> | less **Standard log files to check:** /var/log/audit/audit.log, /var/log/secure *or* /var/log/auth.log, /var/log/messages  
**Systemd logs:** journalctl \_UID=1010 –since yesterday

**Services**

**List all units (systemctl):   
Sort units by state or type:** systemctl --type=service *or*  
 --state=active *or -*–state=running **List all unit files:** systemctl list-unit-files **List service information:** systemctl status <service name *or* unit file name> **List all timers:** systemctl list-timers –all

**Kernel modules**

**List all kernel modules:** lsmod **List module information:** modinfo <mod name without .ko tail> **Load kernel module:** modprobe (it is possible to use insmod but it does not resolve dependencies) **Remove kernel module:** modprobe –r (rmmod is deprecated)

**Dynamic library analysis**

**Listing libraries used:** ldd <path to binary> **Check linker’s conf files/env variables:** cat /etc/ld.so.preload  
 cat /etc/ld.so.cache  
 cat /etc/ld.so.conf  
 cat /etc/ld.so.conf.d/\*  
 env | grep (LD\_PRELOAD|LD\_LIBRARY\_PATH)

**Persistence analysis**

**Cron directories:** /etc/crontab, /etc/cron.\*/, /var/spool/cron, /etc/anacrontab, /var/spool/anacron, /home/$USER/.anacron **Common systemd directories (it is better to start from identifying suspicious units – the paths where you can store them can be very different):  
System level unit dirs:**  
/usr/lib/systemd/system (/lib/systemd/system), /run/systemd/system, /etc/systemd/system  
**User level unit dirs:** /etc/systemd/user, ~/.config/systemd/user, ~/.local/share/systemd/user,  
 /run/system/user, /usr/lib/systemd/user **System scripts:  
User-level persistence:** ~/.bashrc, ~/.profile, ~/.bash\_login, ~/.bash\_profile, ~/.bash\_logout  
**System-wide persistence:** /etc/profile, /etc/profile.d, /etc/bashrc (/etc/bash.bashrc in deb-based distros)

**Evidence acquisition**

**Get ssh key fingerprints:** ssh-keygen –lf <path to public/private key> (passing authorized\_keys file works too)

**Get md5 hash of a file:** md5sum <path to file>  
**Dump process memory:  
Get process memory addresses (for further use with dd):** cat /proc/<pid>/maps | head -1  
 cat /proc/<pid>/maps | grep heap  
**Dump memory (with addresses from the step above):**  
 dd if=/mem bs=1 skip=$((0x08884000)) count=$((0x08970000-0x08884000)) of=/tmp/mem\_dump  
**Get filesystem image:** dd if=/dev/input/DEVICE-HERE of=/dev/OUTPUT/DEVICE-HERE bs=64K conv=noerror,sync  
**Show all logs that contain a string:** grep –rni ‘template.utility.Execute’ /opt/vmware/horizon/logs

**Date/Time stamps**

**Date/time attributes on files:** mtime – dir: last time an entry was added/renamed/removed.  
 file: last time file was written to.

atime – dir: last time dir was searched.  
 file: last time file was read.

ctime – dir|file: last time owner, access perms, hardlink count or any of the timestamps changed (Inode info was edited).

crtime – creation time. Not every filesystem has this attribute and not every forensic utility reads them.

dtime – deletion time.

**Tools**

**Forensic imaging:**

**RAM:  
Dumping process memory with gcore:** gcore <pid>  
**Dumping process memory with gdb:** cat /proc/<pid>/maps  
 gdb –pid <pid>  
 dump memory /root/memdump 0x00621000 0x00622000 (in gdb shell)  
**Dumping volatile memory with AVML:** Download the tool from <https://github.com/microsoft/avml> and run it ./avml memory.dmp  
**Check for hidden processes and malware with unhide/rkhunter/chkrootkit** unhide procall  
 rkhunter –c –-enable all -–disable none  
 chkrootkit