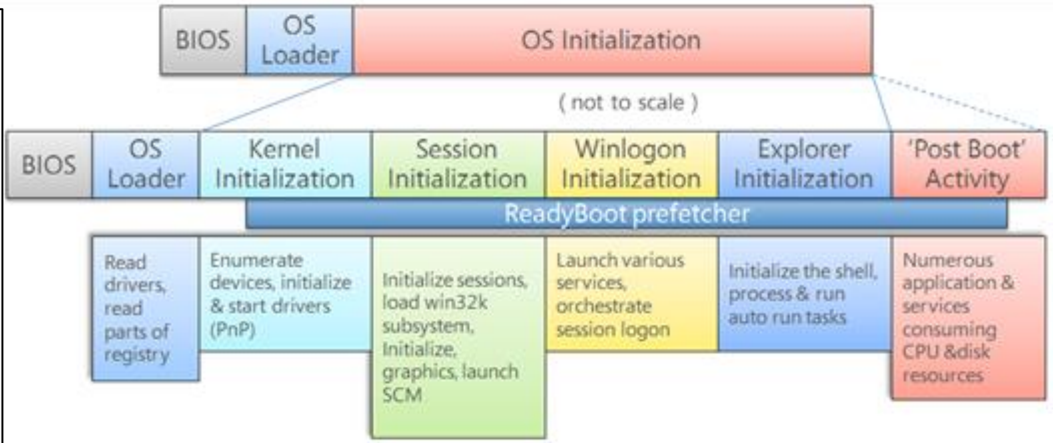
CCTC Windows Page 1

Created by CW2 Jason Fraughton, DET Utah 3/174; Not responsible for missed test questions.

BIOS (WinXP) Pre-Boot
1. Power On Self Test (POST);
2. MBR: Loads boot code
3. Bootcode: Searches partition table for boot sector, loads NTLDR
4. NTLDR: Reads boot.ini to display OS choices, runs NTDETECT.com to query hardware; Stored data from NTDETECT.com in HKLM\Hardware registry key; Kicks off NTOSKRNL.exe and HAL.dll
5. NTOSKRNL.exe: starts SMSS.exe
6. SMSS.exe: Launches Winlogon.exe, CSRSS
7. Winlogon: Starts LSASS (sec policy/logon verify), loads MSGINA, starts SCM, starts logonui.exe, accepts sec attn seq (ct+al+dt)



BIOS
OS Loader
OS Initialization
(not to scale)
BIOS
OS Loader
Kernel Initialization
Session Initialization
Winlogon Initialization
Explorer Initialization
'Post Boot' Activity
ReadyBoot prefetcher
Read drivers, read parts of registry
Enumerate devices, initialize & start drivers (PnP)
Initialize sessions, load win32k subsystem, Initialize graphics, launch SCM
Launch various services, orchestrate session logon
Initialize the shell, process & run auto run tasks
Numerous application & services consuming CPU &disk resources

BIOS (Win7)
1. Power On Self Test (POST)
2. MBR
- First **512 byte** sector on hard disk.
- Reads and loads Volume Boot Record
3. VBR
- Loads bootmgr into memory
4. Bootmgr
- Reads Boot Config Database (BCD)
- Boot menu and memtest
- Calls winload (fresh boot)
- Calls winresume (if resuming)
5. Winload
- Loads NTOSKRNL.exe
- Loads dependencies
- Loads device drivers

EFI/UEFI (req 1GB part – vs. 512 MBR)
- Power On Self Test (POST)
- Runs bootloader out of NVRAM
- Loads BCD (also in NVRAM).
- Bootloader detects hardware
- EFI boot manager gives OS boot menu
- Winload.efi: EFI version of winload
- Requires EFI system partition

NTOSKRNL: SYSTEM; prepares for running native systems; runs smss
HAL.dll: hardware abstraction layer; interfaces driver to kernel
SMSS: session manager subsystem; session 0 loads **Win32k.sys** (kernel subsystem); runs wininit
Wininit: starts Service Control Manager (SCM); starts Local Security Authority SubSystem (LSASS); starts Local Session Manager (LSM)
CSRSS: Client/Server Runtime SubSystem; client side of the win32 subsystem process; thread creation; also handles cmd shell

USERINIT
Last step in Win boot process: 1. load user profile, 2. runs startup, 3. starts explorer.exe

MSGINA: Graphical ID and Auth; Activates user shell; customizable ID and auth. procedures; logon dialog.

Root Registry Keys
HKCU: Current User; indiv user settings
HKU: All accounts on machine; root key containing the ntuser.dat hives for ALL users.
HKCR: Classes Root – file association and COM objects, backward compatibility, and file extension information
HKLM: System related information, SAM, critical boot/kernel functions, 3rd party software, hardware, BCD.dat (boot config)
HKCC: Current Config – Current hardware profile, info gathered at runtime

Logon (Local = SAM; Domain = DC)
Winlogon: coordinates logon and useractivity; launches logonui
Logonui: interactive logon dialog box
Services: loads auto-start drivers and services

Process: executing program
Thread: OS allocated unit to processor time

AUDITPOL: audit policy; seeks abnormalities within system logs

Registry Main Data Value Types
REG_SZ: String
REG_MULTI_SZ (end \0): MultiString
REG_BINARY: Binary
REG_DWORD: 32B INT
REG_QWORD: 64B INT
REG_LINK (symbolic): Link:

Windows Resource Protection
Prevents changes to DLLs, SYS and EXEs; ensures cached copy to boot

How to Modify Permissions
GUI: rt click + props, sec tab
CMD Line: icacls.exe
Powershell: get-acl [resource]|Format-List
Sysinternals: accesschk [resource]

Forensically Relevant Keys: Run | RunOnce | Services | APPINT_DLL (current logon session) | Shell Ext (Startup folder) | Scripts

Auditable Events (seen in MMC, get-eventlot):
User accounts/permissions
Failed Logon attempts

SAS: Secure Attention Sequence
Tells system you want to authenticate (CTRL+ALT+DEL); kernel detects key combo to initiate trusted login process

GPO: GPEDIT | REG cmd

Auditpol: Used to establish procedures, look for abnormalities w/ system logs

Registry Defined: DB of system configs, contains configs, settings for entire OS, users

Viewing Event Logs
Wevutil | Get-EventLog
wmic nteventlog list brief

Compare Reg Keys
Get-child-item | PROCMON
Get-ItemPropery

Currently Logged On
LOGONSESSION, PSLOGGEDON

Static: Strings, IDA Pro
Dynamic: Reg Shot, Wireshark

Piping in PS: output from first cmd is input for the next cmd

Get-Member: shows object type, props, methods of object
Group Commands: net group
net localgroup | get-localgroup

Objects: PS data structure; properties = data methods = actions taken on those props

? Where-Object
% ForEach-Object
$_ Variable placeholder for current value in pipe

Users
wmic useraccount: local accounts w/props
net user: just lists accounts
get-localuser: name, enabled, desc

Groups
net localgroup: just lists all aliases
get-localgroup: name, desc

System
set: all current env. Vars (kinda messy)
systeminfo: props, hotfixes, nics (cleaner)
get-hotfix: patches (srce, desc, id, inst dte)
get-computerinfo: comprehensive

Processes *PROCMON: GUI
tasklist: more like taskmgr
wmic process list brief: more raw; w/ handle, thread, etc
get-process: like wmic process (w/ handles)
taskmgr: gui
gwmi win32_process: huge list of wmi proc.
*pslist –t: shows a tree

Services
wmic service list brief: like "services"
sc query: service control (change/query)
net start: svcs open on startup
*psservices: config'ed svcs on local sys

Network
ipconfig:
netstat: local, foreign sockets and state
 -a: listening ports | -b: processes
net view: member net only (brief)
wmic nicconfig: all nics (lots of info)
arp –a: IP, MAC and type (stat or dyn)
Get-NetIPAddress: all sys IPs (link, dhcp, etc)
Get-NetIPConfiguration: adapters (vmw, eth)
*tcpview: sysinternals tool (think Wireshark)
**=ext cmds*

Heuristics: zero-day; >false positives (logic)
Signature: previous ident. attacks; >updates; easy to bypass

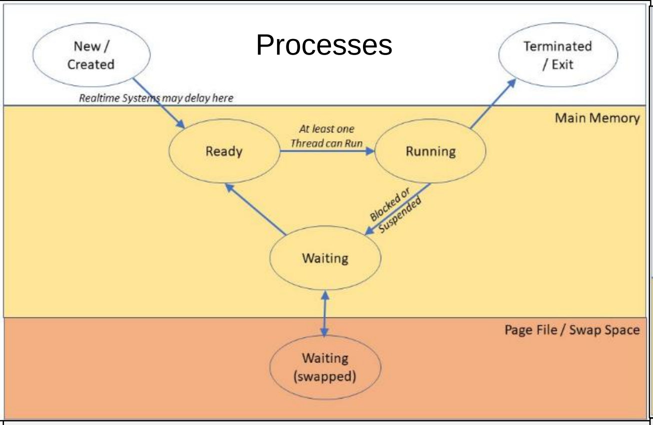
Opnotes: taken during mission, fed back to reports with findings; sys baseline; dev TTPs

NetBIOS:
137: Name Registration
138: Datagram Messaging
139: NBT (NetBIOS over TCP)

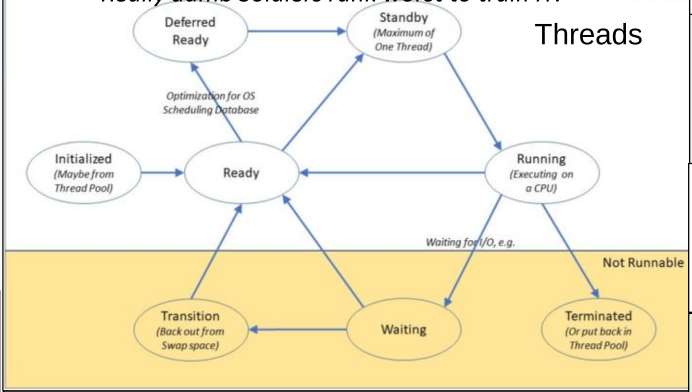
Process: executing program
Thread: several make up a process

CCTC Windows Page 2

Incident Response: Please include chocolate in every recipe.



"Really dumb Soldiers rank worst to train IT."


Runnable
Threads
Deferred Ready
Standby (Maximum of One Thread)
Optimization for OS Scheduling Database
Initialized (Maybe from Thread Pool)
Ready
Running (Executing on a CPU)
Waiting for I/O, e.g.
Not Runnable
Transition (Back out from Swap space)
Waiting
Terminated (Or put back in Thread Pool)

Process States
New/Created
-Open the file (.exe)
-Create initial thread
-Pass to kernel32.dll to check permissions
-Pass to csrss, build structure, spawns first sub-thread, inserts into windows subsystem-wide proc list
-Starts execution of initial thread
-For real-time systems, processes may be held in "New State" to avoid contention
-Otherwise, move to "Ready State" automatically
Running: Process currently being executed (one or more threads executing)
Ready: Process is ready to execute when given the opportunity (CPU Time)
Waiting: Process can't execute until some event occurs (I/O Read)
Terminated/Exit: Termination of a process due to a halt or abort

SYSINTERNAL TOOLS
PROCMON: view, monitor, filter processes/registry
PROCEXP: handles, dlls processes have opened/loaded
ACCESSCHK: modify permissions
AUTORUN: check autorun registry locations
HANDLE: -p <process name>
ROOTKIT REVEALER: Doesn't work on 64b
STRINGS: see what DLLs, func's, headers might reveal in output
TCPVIEW: net traffic
TCPVCON: built-in netstat utility
PSLIST: processes in tree format

Add Registry Key
reg add hklm\... /v <name> /t <data type> /d <data>

User Account Control: Prevents unauthorized changes to system; requires limited elevated prompt to make them

Thread States
"Really dumb Soldiers rank worst to train IT."
Ready: Waiting for execution, in priority pool
Deferred Ready: Selected to run, but not yet executed. Optimization for scheduling database
Standby: Next thread to run, only one per processor per system
Running: A thread currently running on a processor
Waiting: A period of inactivity while waiting for an event
Transition: Ready for execution, but paging needed to bring stack back into memory
Terminated: Finished execution, heading for deallocation in most cases
Initialized: Thread is being created

Firewall Rule Setting Methods
GUI
NETSH AdvFirewall Cmd
Manipulate Registry

SMB Server Message Block Protocol; used for sharing access to resources.
2.1: added MTU size
3: Uses AES encryption

IA: prevents unauthorized access, use, disclosure, disruption, mod, inspection, recordings or destruction of info

Mailslot: one-way IPC; apps store messages in them; between client and server

Rights Admin 1. ICACLS
2. ACCESSCHK | 3. Right click(y)

PS Syntax: Object.<Property Name> or Object.<Method>(Args)

Socket
One endpoint of 2-way comm (IP+Port)

Winlogon
LSASS: sec policy
MSGINA: XP logon
SCM: sessn ctrl mgr
LOGINUI: user, p/w

Logging Triggers
Account changes
Logon attempts
Resource access
System file changes

Incident Response
1. Preparation
2. Identification
3. Containment
4. Investigation
5. Eradication
6. Recovery

Paging is…
RAM overcommit, page write to disk

MDMP (CCTC Version)
1. Mission Receipt
2. Mission Analysis
3. COA Develop
4. COA Compare
5. COA Approval
6. Conduct Mission
7. AAR/Lessons

Collected Data Order of Volatility
1. Cache
2. Route table, arp cache, process tbl
3. Temp file systems
4. Disk, other storage media
5. Remote logging, mon. database
6. Physical config, net topology
7. Archival media

Setting a Variable Example
$host = "$env:COMPUTERNAME"

Setting a Firewall Rule Example
New-NetFirewallRule -DisplayName "Block 80" -Direction Outbound -LocalPort 80 -Protocol TCP -Action Block -AsJob

Setting Execution Policy Example
Set-ExecutionPolicy Unrestricted -Scope CurrentUser
**Protecting you from yourself, and protecting self from attacks.*

Registry Keys – Forensically Relevant
-HKCR (File extensions)
- HKLM and HKCU have Software keys with default settings
- HKLM\system\CurrentControlSet

Looking for PS Field Example:
Get-Process | Get-Member Findstr /i pri

Virus: user interaction to replicate
Worm: no user interaction
Trojan: hidden w/in a legit program; not usually self-replicating
Malicious Mobile Code: xmit from remote to local host; w/o user int.
Blended Attack: multiple methods

Backdoor: illegit access; user unaware
Remote Access Tool (RAT): provides remote command/control (C2)
Rootkit: ONLY used to hide things; DOES NOT provide access or C2 alone; loads with bootloader
Keylogger: records keyboard usage
Botnet Client: remote admin / C2 of botnet
Spyware: monitors behavior of user
Adware: paid for ads to infected users
Ransomware: blocks resource access; requires victim to pay

Exec Sum: 10k view for less tech indiv.
Tech Sum: much more in depth