System Examination

Linux Systems Security

Availability

- Computer systems exist to satisfy an expectation of service
- Expectations have multiple facets
  - Functionality
  - Features
  - Consistency
  - Responsiveness
  - Support and recovery
- Documentation and training usually define these, deviations from the norm can be indicative of system compromise
- Examining running systems identifies deviations, users do it constantly as a byproduct of using the systems

Stability

- Stability refers to the continued, consistent provision of expected services
- Misconfigurations and bugs can introduce instability
- Attacks can produce instabilities that look like bugs or misconfigurations, successful intruders may alter configurations
- Normal operations and configurations must be clearly documented and well understood, to be able to identify abnormal situations
- Instabilities are commonly flagged by users, but can also be identified by tools, logs, and reports

Evidence of Compromise

- Unexpected behaviour requires investigation e.g. resource usage spikes (up or down), inconsistent system responsiveness or responses, i/o irregularities
- User problem reporting identifies symptoms, but not always causes
- Monitoring systems can alert you to anomalous behaviour - sometimes only external monitoring will show anything, depending on the cause
- Log files can be reviewed or analyzed for additional symptoms or root cause indicators
- Simple examinations of critical services and files can provide evidence of compromise

Documenting Normal

- Manuals and business policy & procedure documents can define expected system operation
- State information such as resource consumption and usage, and the rates at which these things are changing are things to consider including
- Recording these things removes dependency on the guru

System State - Static

- System components (configuration and program files) should only change in a planned way
- The state of these files can be compared to their expected state
- rpm and debsums provide tools to check if program files have been modified other than by package installation and update procedures

rpm -qV packagename, dpkg -V packagename

debsums -s [packagename]

- Neither can do much which is useful with generated files or configuration files, create your own methods, scripts for this
- *debsums* does a different check from the *rpm* command which is more thorough
- It is not uncommon for systems to have packages installed using more than one package management system (terrible very bad idea, IMHO), beware of snaps - read-only mounts are not a guarantee of integrity

State Verification - Static

- Check for setuid/setgid files not belonging to installed packages
- Check for files in user directories not owned by those users
- Check for files in system directories not belonging to installed packages
- Check for files owned by system service daemons not belonging to installed packages
- Pay particular attention to files and directories whose names start with a dot, especially if the files are executable
- This is a good task for a script!

File Integrity Tools

- rkhunter is a tool which can look for a number of types of malicious files although development is sporadic so it shouldn’t be your only tool
- Tripwire has been around for a long time and is a general purpose tool to identify file changes
- Tripwire is available in both free and enterprise paid versions and has a GUI
- Advanced Intrusion Detection Environment (AIDE) provides for comprehensive checks of your file stores to identify changes and was intended to be a replacement for Tripwire before Tripwire went commercial

System State - Dynamic

- Dynamic state is
  - use of resources (cpu, memory, storage, network) - performance measurement
  - current access summary (who is on, what are they doing) - user access
  - service(s) status (running, stopped, degraded or normal) - service evaluation
  - external views of traffic flow (who/where from/to, what ports/services, volume, path taken) - unusual activity flagging
- Check running system configuration with respect to time, timezone, resolver, network config, storage config, things that get set dynamically at boot or during normal operation
- /proc filesystem provides lots of raw dynamic information when you need to dig down

Performance Measurement

- Use of resources
  - CPU/memory
    - top, htop can be used for simple overview of running system, or use a more sophisticated toolset like glances
    - ps, pidstat, vmstat, memstat, mpstat, free, pmap, pstree can be used to drill down and investigate things not running in a normal state
    - sar can be used to view historical data for comparison purposes, enable in /etc/default/sysstat and restart service to start data collection
    - ac (from psacct or acct) can also show summarized past usage info

Files and Storage

- iotop, df, du, find can be used to check current activity and usage and detail it as necessary
- mount/umount, automounting is often configured to permit end-user mounting of filesystems and should be investigated to ensure any user-mounted filesystems are nosetuid, and probably also noexec
- swapon (*free* includes swap usage), lsof, iostat can be used to drill down to the details
- world-writable directories, filesystem type-based limitations are potential attack vectors
- Often, the worst thing that happens to a Linux server is that it gets misused to host Windows malware - clamav, bitdefender, and others can be used to actively scan your file stores for Windows malware

Performance Overview Tools

- Glances is a tool that shows a summarized subset of information about the running system
- KDE and Gnome have process monitoring tools that allow graphical process and performance exploration
- Everybody and their sister writes custom tools for their own environment, many get published under GPL because they start with some other piece of GPL code
- Check out monit, monitorix, nagios, nmon, collectl, web-vmstat for examples of real-time enhanced performance monitoring tools
- No matter which tool(s) you use to become aware of anomalous conditions, there are plenty of commands to help dig into what is going on, and don't forget about /proc which gives you a comprehensive view of what is going on inside your machine

Performance Tools References

- There are so many tools available, lists of tools are also common

http://www.tecmint.com/command-line-tools-to-monitor-linux-performance/

http://www.tecmint.com/linux-performance-monitoring-tools/

http://www.cyberciti.biz/tips/top-linux-monitoring-tools.html

https://blog.serverdensity.com/80-linux-monitoring-tools-know/

User Access

- Unix users can be tracked or observed by a number of tools
  - who, whoami, id, w
  - last, lastcomm, history
  - various more comprehensive auditing tools
  - Most of the time, if you see users on that should not be on, or shouldn't exist, you have already been broken into

Services Evaluation

- Services can be checked in a number of simple ways
  - systemctl, service, ps, netstat, nmap
  - telnet, per-service access tools - connecting to a service is the only way to know for sure that it is responding normally
  - log entries - helpful for figuring out what went wrong (past tense) or just knowing something went wrong when all appears normal otherwise
- Various services have more sophisticated built-in status inquiry mechanisms
  - apache has server-status module
  - mysql internal status commands and queries
  - cups lpadmin and web interface toolset
  - Some services are complex enough that there is software available to evaluate its status and manage the service - e.g. phpmyadmin for mysql-compatible database servers

Network Examination

- Network
  - ntopng, iptraf-ng, nethogs, iftop, arpwatch
  - netstat, route, ss
  - ifconfig, ethtool, ip
  - resolver configuration, might be a static verification
  - ping, traceroute, arp, nslookup, tcpdump, wireshark can all be used to drill down when investigating anomalies

Log Entries

- Logs are typically kept in /var/log as a default
- Programs may directly write their own log files, or they may use the syslog service
- Syslog can be configured to manage messages in custom ways, including sending them to a log server
- Each program makes its own decisions about what to log, when to log it, and how usefully to describe whatever is being logged
- Beware of sensitive information appearing in log files, they are not encrypted
- Refer to the man pages or user guides for each service to see what is logged and how