LSE

# UNITS cont'd - WORKSHEET

## UNITS DP1

- Read the rockylinux guide chapter 13

1. What do we understand about the process

2. What new things did you learn

3. What are the DNF plugins near the bottom
   ie
   a. versionlock plugin - locks to a certain version helps to keep related packages in sync with each other
      helps to keep multiple
4. What is an EPEL * companies like AWS never support EPEL so if you use EPEL pkgs then some companies will deny paid support in enterprise
   a. Why do you need to consider this when using one?

## UNITS DP 2

- Google search "patching enterprise Linux"

1. What blogs did you find listing steps / checklists

2. After review, how does enterprise differ from one off / update now
   a. What are the major considerations
   b. What may be major roadblocks

# UNIT 5 Lab

Stig viewer
└ APACHE 2.4 Unix Server
1. Search "tls" in Stigs
2. Install httpd on hammer
   #systemctl stop nmcbent
   #dnf install -y httpd
   #systemctl start httpd
3. Check STIG V-214234
   a. problem?  b. fix?
   c. type of control? TECHNICAL DETECTIVE CONTROL
4. Check STIG V-214248
   a. problem?  b. fix?
   c. type of control?

BUILDING REPOS

1. Start out by removing all your active repos
   #cd /etc/yum.repos.d
   #mkdir old_archive
   #mv *.repo old_archive
   #dnf repolist
2. Mount the local repository and make a local repo
   #mount -o loop /lab_work/repos_and*/Rocky*.iso /mnt
   #df -h
   #ls -l /mnt
   #touch /etc/yum.repos.d/rocky9.repo

cont'd

```
# Vi /etc/yum.repos.d/rocky9.repo
    [BaseOS]
    name=
    ...
    baseurl=file:///mnt/BaseOS/
    ...
    [AppStream]
    ...
    baseurl=file:///mnt/AppStream/
    gpgkey=...
# chmod 644 /etc/yum.repos.d/rocky9.repo
#dnf repolist
#dnf clean all
```

3. Test the local repositor

```
#dnf repolist
#dnf --disablerepo="*" --enablerepo="AppStream" list available
#dnf --disablerepo="*" --enablerepo="AppStream" list available | nl
#dnf --disablerepo="*" --enablerepo="BaseOS" list available
#dnf  "  "  "  list available | nl
```

4 Try to install something

```
#dnf --disablerepo="*" --enablerepo="BaseOS AppStream" install gimp
```

5. Share out the local repository for your internal systems

```
#rpm -qa | grep -i httpd
#systemctl status httpd
#ss -ntulp | grep 80
#lsof -i :80
```

```
#cd /etc/httpd/conf.d
#vi repos.conf
    <Directory "/mnt">
        Options Indexes FollowSymLinks
    :
    Alias /repos /mnt
    <Location /repo>
    :
    </Location>
#systemctl restart httpd
#vi /etc/yum.repos.d/rocky9.repo
    [BaseOS]
    :
    baseurl=http://____/repo/BaseOS/
    :
    baseurl=http://____/repo/AppStream/
    :
#dnf clean all
#dnf repolist
```

ENTERPRISE PATCHING

- custom facts
- prechecks
- reboot task

# Ansible in the Enterprise (killercoda.com/het-tanis)

Clone the git of HPC_Deploy

```
#git clone https://github.com/het-tanis/HPC_Deploy.git
#cd HPC_Deploy
#ls
#cat hosts
```

```
[servers]
_____ type=client app=web
_____ type=server app=db
```

```
#ansible-playbook -v -i /root/HPC_Deploy/hosts
  ↳ 03_package_update_or_install.yml --extra-vars "action=install"
#cat 03_package_update_or_install.yaml
```

```
- hosts: all
  gather_facts: true
  vars:
  tasks:
  roles:
  - { role: packages_install, when: action == "install" }
  - { role: fact_push, when: action == "install" }
  - { role: packages_update, when: action == "update" }
```

```
#ls roles/fact_push/files
#grep database db_patching.fact

#grep webserver web_patching.fact
#ls ../tasks/main.yml
```

#cat main.yml

```
- name: Create the facts directory
  file:
    path: /etc/ansible/facts.d
    state: directory
- name: Push the db_patching.fact
  copy:
    src: db_patching.fact
    dest: "/etc/ansible/facts.d/patching.fact"
  when: '"db" in app'
- name: Push the web_patching.fact
  copy:
    src: web_patching.fact
    dest: "/etc/ansible/facts.d/patching.fact"
  when: '"web" in app'
```

#cat /etc/ansible/facts.d/patching.fact

#cat /root/HPC_Deploy/04_enterprise_patching.yml

#cat /root/HPC_Deploy/roles/precheck/tasks/main.yml

#cat /root/HPC_Deploy/roles/fact_push/tasks/main.yml

#cat /root/HPC_Deploy/roles/patch/tasks/main.yml

#cat /root/HPC_Deploy/roles/reboot/tasks/main.yml

# UNIT 6

Logging & how logging works

LST WK TOPIC 1 : Building Repos

LST WK TOPIC 2 : Enterprise Patching *still more coming in future*

We deal in truth - So we need to earn the company's trust

Enterprise patching is about making sure all servers are in sync
↳ not just updated but on the same stable release

Logs & Parsing
↳ An immutable record of an action that occured/or tried to occur
- Immutable - write once, read many ; doesn't change ?

Timestamps, synchronization

Goal #1 We store logs, Get them off the server in real time is essential
* we need logs to determine if an outage was malicious vs accidental

Parsing logs with command line tools vs Querying logs with specific tools

Log usage : Review

Troubleshooting - logs provide valuable clues about what happened

Security Auditing - track user activity, to identify security breaches

Log Usage : Checkpointing

Performance Monitoring - insights into system performance & resource utilization

Debugging Applications - Developers rely on logs to debug

*Cisco will crash if you enter debug all* It will crash the router

RFC 3164 BSD Syslog - Berkeley Software Distribution

RFC 5424 IETF Syslog - newer

BSD is a simple format

IETF provides structured data:

*Cisco devices, etc use IETF

.yml

Systemd Journal
↳ It's in a binary format
↳ We need the command #journalctl to read the logs
#journalctl
*most Linux distros are using systemd

Log Storage & Rotation
/etc/logrotate.d
/etc/logrotate.conf
Log rotation is crucial for stability. Filling up logs can crashs

Sending logs off *TOP FOUR TYPES OF ARCHITECTURES
1. rsyslog (cheap, free) FORWARDING & COLLECTION
2. Elk Stack, Splunk, Graylog, Loki (AGENTS & a central/cell Platform)
3. Kafka, RabbitMQ (Message Queues) Super highly redundant/complex
4. VPC flowlogs (cloud native)

PARSING LOGS WITH TEXT TOOLS
- Understand the log format
- Start Simple
- Use Regex RegularExpressions with grep & sed
- Iterate & Refine
- Save useful commands

Special tools
- Splunk
- LogQL
- ELK (Elastrosearch, Logstash, Kibana)
- Graylog

Grafana.com/docs/loki/latest/query/analyzer

Libraries

(Splunk is good at finding anamolies on a disk that indicate that the disk is about to die)

UNIT6 DP1 - Julia Evans Blog

UNIT6 DP2 - sre.google

#grafana default user/passwd admin/admin

Kafka is used because it's highly reliable
↳ very valuable on a resume, makes resume shine
↳ use this lab to as

landscape.cncf.io/guide#

Review
A log's value is to be used in troubleshooting,
logs need to standardized & have accurate timestamps
Rotating logs, storing logs off server/centralized,
SPLUNK - is just expensive but works well heavy forwarder forwards
LogQL - free
ELK stack of Elasticsearch, Logstash, Kibana - free but
EXPERTISE -
↳ Splunk training is free for Veterans ☹

# UNIT 6

LAST WK1 TOPIC 1 - BUILDING REPO'S
TOPIC 2 - ENTERPRISE PATCHING

WHY DO WE CREATE A REPO LOCALLY
↳ TO BE ABLE TO UPDATE SOFTWARE BUT CONTROL VERSIONS USED

HOW DO WE CREATE A REPO
↳ LOCAL VIA FILE SHARED OFF A MOUNTED ISO ON THE NETWORK
↳ ALSO VIA HTTPD DAEMON /etc/yum/repos.d

ENTERPRISE PATCHING VS AUTO UPDATE DOWNLOAD PATCHING
↳ PRE CHECKS, PATCHING, AFTER PATCH VALIDATION OF APPS

## THIS WEEK - LOGS

TOPIC 1 - STORING LOGS - WHAT, WHERE, HOW, HOW GATHER
TOPIC 2 - PARSING LOGS - HOW DO WE LOOK THROUGH LOGS

WHAT IS A LOG
↳ IT IS AN IMMUTABLE RECORD OF AN ACTION THAT OCCURED
↳ EVEN THE ATTEMPT TO DO SOMETHING SHOULD BE LOGGED

HOW DO WE USE LOGS
↳ TO INVESTIGATE SECURITY INCIDENTS
↳ WE HAVE TO GET LOGS OFF A SERVER SO THEY AREN'T ALTERED
↳ ALSO, IF THE SERVER GOES DOWN THEN OFFLINE LOGS ALLOW US TO TROUBLE SHOOT

LOG USAGE
TROUBLE SHOOTING
SECURITY AUDITING
PERFOMANCE MONITORING
DEBBUGGING APPLICATIONS

SYSTEMD JOURNAL

RFC 3164 BSD - Syslog *doesn't include the year in timestamps
RFC 5424 IETF Syslog
ISO 8601 - logs must be timestamped IETF
LOGS MUST HAVE SEVERITY CATEGORIES
- CATEGORIES e.g. DEBUG, INFO, NOTICE, WARNING, ERR, CRIT ALERT, EMERG

3164 <DATE> <TIME> <HOSTNAME> <PROCESS> <PID> ...log...

IETF 5424 <DATE> <TIME> <HOSTNAME> <PROCESS> <PID> ...msg...

SYSTEM D JOURNAL
* LOGS ARE IN A STRUCTURED BINARY FORMAT
* WE USE JOURNALCTL TO VIEW LOGS

<__REALTIME_TIMESTAMP> <_HOSTNAME> <_COMM> <_PID> <MESSAGE>

LOG ROTATION
/etc/logrotate.conf → specifies GLOBAL generic/default logrotate settings
/etc/logrotate.d/ → these are specific logrotate setting for specific apps
* log rotation helps us maintain space & organize for investigations (e.g. incident response usually is just the last few (3 days)

COMMON ARCHITECTURE FOR SENDING LOGS OFF THE SERVER
1. RSYSLOG - FORWARDING FROM LOCAL DAEMON ON EVERY SERVER TO A COLLECTION SVR
   - ① UDP ② SECURE VIA CERTS ENCRYPTED - NATIVE & ALREADY BUILT IN
2. LOG AGG W/ AGENTS - PUT AGENTS ON MACHINE TO FORWARD LOGS TO COLLECTN SVR
   ① ELK STACK ② SPLUNK proprietary expensive ③ GRAYLOG open source ④ LOKI w/ GRAFANA FROM TAIL
3. MESSAGE QUEUES - ① EVENT BUS / EVENT DRIVEN ② CAN REPLAY IN ORDER
   ① KAFKA ② RABBIT MQ ③ IBM MQ / MQ SERIES
4. CLOUD NATIVE LOGGING SVCS ① EVENT BRIDGE ② ALREADY BUILT IN

WHY CHOOSE ONE OVER THE OTHER?

- SCALE -
- RELIABILITY -
- REAL TIME ANALYSIS
- BUDGET / • EXPERTISE / • COMPLIANCE REQUIREMENTS / • EXISTING INFRA

PARSING LOGS WITH TEXT TOOLS

#grep rhost secure* | awk '{print $NF-3}' | sort | uniq -c

SPLUNK - BUILDS PROFILES

I used SPLUNK & ARCSIGHT SIEM at Citi

WHEN TO USE SPECIAL TOOLS

- Large Log Volumes - Complex Log Formats
- Need for Real-time Analysis & Alerting
- Requirement for Visualization & Dashboards
- Security Analysis - Compliance Requirements

LONG LAB

LSE

# UNIT 7 Live

Last Week - Storing Logs • Parsing Logs • Security standpoint
↳ Get the logs off the system • Make sure they dont change
• Immutable record of history. 2 RFC structures of logs
• Store logs in a centralized location • Rotate logs for space & compression
• Recent logs are most actionable & faster to search

This week - Monitoring • tracing signals • Alerting
Alerts come out of a system to know info, needs to be relevant

# Webhook Security Key → same as hammer lab Password

Overview - Telemetry • Trace • Logging
Prometheus has to scrape & grab info or it's lost forever
Alerting - send messages off the system to a control center
• Alert Manager
• Grafana

Types of data to monitor: ① Logs ② Metrics ③ Traces: data moving @

SysAdmin -

SysEngineer - System performance: HIGH CPU • Connections • User Login
↳ Watching User login is important, bcuz, a compromised user is a big risk

# GIT is a security Risk - Keys can be compromised from Past County

Monitoring - FORENSICS - after action reviews — LOW URGENCY
- INCIDENT - REAL TIME SOLVING OF A COMPROMISED SYS — TIME CRITICAL

Push -vs- Pull : Push is not preferred • e.g. Telegraf, Promtail
↳ PULL - IS PREFERED • aka SCRAPE •
e.g. Prometheus - using http • If Prometheus doesn't scrape THEN it's LOST FOREVER

* Timescale DB or InFluxDB ?? AWS?

Concept of Alerting - Paging out · Sending info out

* Slack (free version saves only 1 month)

PAST ALERTS - ANALYZING LOGS

CURRENT ALERTS - ① HIGH or LOW THRESHOLD being crossed
② TIME LIMIT IS EXCEEDED ③ SERVICE or SERVER stops RESPONDING

FUTURE (predictive)

When do we alert

- Security breaches: Suspicious activity · Malware infections
- System failures: Server downtime · Network outages
- Performance issues: High resource usage (CPU, memory · Slow response

* Kubernetes has a lot of warnings (CKAD cert?)

V7 DP1

V7 DP2 - a. what is litmus test for page out b. over vs under c.

Definitions → Look up SIX SIGMA · IDS only sends alerts (watches only) vs IPS changes things (has rules)

IDS - watches only · only sends alerts

IPS - have rules that can take action to protect the system
↳ It will drop the gates, it will take action

Wazuh - Indexing & storage as JSON data?

— x —

LSE

# UNIT 8 live

TOPIC 1/2- What do we monitor; Metrics, Telemetry, logging
- Telemetry (Metrics) - Push or allow them to be scraped (pull)
- Trace & Logging & Alerting & Time Windows (Critical during weekend)

Lifecycle of a system: The deployment aspect
Once built into a baseline system... the config starts to drift....
How do we deal with configuration drift from one machine vs another machine
Operational pratices vs Technical practices.
We do 2 things: ① Check things ② Fix things
Security is the enemy of productivity.
Fixing things is always destructive - fixing wipes out others changes
Configuration management
- Config Item: hardware, software, documentation
- Baseline:
- Change Control: document every change
:
- Config AUDIT: Making sure everything is where it should be

CMMB: RACI...

CONSISTENCY: Ensure that system is built & operates as originally intended
TRACEABILITY:
ACCOUNTABILITY: DEFINE WHO IS RESPONSIBLE FOR CHANGES
CONTROL:
EFFICIENCY:

Use md5sum or sha256sum → to check for changes
↳ Run a hash against a file, then come back a year later, & then a new hash
↳ Check to see if the results changed

Version control
Infra as Code
* Orchestration - Various ansible based tools
Change tracking
Compliance & auditing

— x —

version control Git for version control
Puppet, Chef & Ansible (still requires a separate orchestrator)
Docker, Kubernetes
Terraform

— x —

V8DP1
V8DP2 - Show where you can be useful

Install aide pkg for the lab

# UNIT 7 recording

## What do we monitor?

TELEMETRY (metrics)
- COMPUTE, MEMORY, DISK, NETWORK

TRACE
- what the flow of a process as it runs on a system
- trace data: the flow of data (tracing data as it flows through (e.g. Query in a browser => front end/loadbalancer => Middleware svr => DB...)

LOGGING

How do we monitor?
We set up apps to catch logs from various systems

What is alerting?
Escalation when a threshold is reached
- send msg's outside of the system

How do we alert?
- Alert manager → Discord (Prom/G lab example)
- Grafana → Discord (Prom/G lab example)

When do we alert
- Priority: Severity levels WARN vs CRIT, etc
- Windows of time: Time of day determines whether to alert

---

WHAT TYPES OF DATA DO WE WANT TO MONITOR? CARE?,

LOGS
- logs are an immutable record of what has happened on system
- *dont change the original logs*

METRICS
- a set of numbers used to assess: resource utilization
  - ↳ system performance, application performance,
  - Utilization (0-100%), Saturation (Queue lengths), Error metrics (%)

TRACES — Tracking how data is moving through our system. Checkpoints of processes as data moves between nodes

FROM THE SysAdmin Standpoint

When we start having issues... for every Resource... Check

Utilization , Saturation, Errors

↳ This tells us how healthy a system is

LSE

# UNIT 9 Live

#ssh-keygen
↳ test
↳ password
↳ password
⇒ Output "They key fingerprint is:
SHA256 ----

#ls -l
test
test.pub

#ssh-keygen -l -f test
⇒ Output "3072 SHA256---

#ssh-keygen
⇒ Output "3072 SHA256...

#ssh-keygen -y -e -f test
⇒ Output will be the public key
---BEGIN SSH2 PUBLIC KEY---
Comment: "307

In the above, we used our private key, to REMIND US of the public key

Zero trust:

vs

Certificate: Public keys that trusted

*Read Up the details on ZT - Zero Trust

↳ ZTA

X.509

---

In a zero trust environment

↳ systems only accept connections from trusted servers / clients

*even the webserver will not

# apt -y install

# certool --generate-privkey --outfile ____.pem

# certtool --generate-self-signed --load-privkey ____ --outfile ____

# certtool --generate-privkey --outfile key.pem

# certtool --generate-request --load-privkey ____ --outfile ____

# certtool --generate-certificate --load-request ____.pem --outfile ____

↳ --load-ca-certificate ca.pem --load-ca-privkey ca-key.pem

# cert

# certtool --certificate-info --infile ____.pem

# scp

vi -x webhook
↳ password is SecLab12#$5

* The above file is encrypted so we need to use (vi -x <filename>)

BlowFish → TwoFish
Encryption

---

logSeQ
↳ /draw
↳ connects right to excalidraw

# UNIT-live

Next meeting is June 14 (June 7 is a day off)
Final meeting on June 7
U10DP1 – I need to record terms NOW
↳ The DP is about terms I heard during the live
U10DP2 – Eg. Stig vs how I built vs Controls
- Constantly think about vulnerability... List what u include
↳ List things you feel comfortable talking about

Last week Key Management
Last week TLS Encryption
You can trust your own CA • Deploy to internal browsers to trust the internal CA

Topic 1 – Journal of topics:
Topic 2 – Proof of skills:
Security+ Exam Topics by CompTIA
Simple Communication Model
① Sender ② Receiver ③ Message ④ ⑤ Noise
What is security? CIA Triad
What is a compliance standard
PCI DSS – 12 steps. Attestation..
What is authority? Authority only matters if it is respected
What are controls? Tech vs Admin vs Physical

| | |
|---|---|
| Preventative | Corrective |
| Deterrent – Signs to start | Compensating |
| Detective – | |
| Directive – Telling you what to do | |

...

CIA (Confidentiality, Integrity, Availability)

Non repudiation (I know I was talking to you)

AAA (Authentication, Auth, Accounting

...

2.5 Segmentation, Access control,
Isolation, Patching
HIPS Host based intrusion prevention
Installing endpoint protection

3.3 Data types: Regulated, Trade secret, Intellectual property
We as sysadmin are data custodians not data owners
Data at rest AES256 minimum standard
Data in transit - must use TLS
Data in use - Masking
Data sovereignty - protecting specific country's citizens data
IPV6 - can't hide location from this (IPV4 offers no protection)
Masking
Masking - hide some of the bits
Obfuscation

5.1 Summarize elements of effective security governance
Policies - high level behavior limits / rules
Standards - Encryption (e.g. the system may require FIPS ##)
Procedures - They may specify certain checks then when to escalate
Monitoring & Revision of (docs, etc.)
Roles & responsibilities: We are custodians

5.2 Risk Management NIST

- Risk assessment: Adhoc, Recurring, One time, Continuous
- Risk analysis: Qualitative, Quantitative
  - SLE
  - ALE
  - ARO
  - Probability 0 to 1 or 0% to 99.9%
  - Likelihood, Exposure Factor, Impact
- Risk register Responsible Accountable Consulted Informed
- Risk tolerance
- Risk appetite: Some time you do nothing (build replacement)
- Risk management Strategies
- Risk Reporting
- Business impact analysis
  - RTO
  - RPO
  - MTTR (if you are not monitoring properly you miss problems)
  - MTBF (means we need parts & spares

UNIT 2

sysctl, dns security, ngrep, /etc/nsswitch.conf

STIG RHEL 9, firewalld, ansible STIG automation

3.2 Secure Enterprise Infrastructure

Failure mode: fail-open, fail-closed

Device attribute: inline vs. tap/monitor

fail-open = human life/protection of human life is TOP

Load balancers aka a reverse proxy (they sit in front)

Firewalls operate up to layer 7

- WAF
- UTM
- Layer 4 firewalls are great for speed
  - they are good at end to end communications
  - they watch half open

Secure communication/access

- VPN
- TLS - Layer 4
- IPSec

4.5 Enterprise capabilities to enhance security

- UBA
- XDR
- EDR
- NAC
- DLP
- SPF
- DKIM
- DMARC

*Network Canary - a device that sits just to tell you that someone scanned it

LSG

# UNIT 3

4.6 IDENTITY & ACCESS MANAGEMENT
Never delete accounts of old admins
Federation
SSO - Single Sign On ; OAUTH 2 or Shibboleth
LDAP, OAuth, SAML
Access controls: Mandatory, Discretionary Role based, Rule based
Multifactor Auth: Something You know, Something You have,
Password concepts
Privileged Access

## UNIT 4

Jailing, Bastions, Air gapped systems
Tectia client - secure file transfer, remote access

## UNIT 5

Package Integrity, trusted repos v/s internal repos
CVSS 7 or higher should be pushed immediately

4.2 Security Implications
Enumeration - just counting through things
Good policies & Procedures will get management on your side

4.3 Vulnerability management
Dast vs Sast
Threat feed against CVSS
Advanced persistent threats are always there

# UNIT 6

RFC 3164 & 5424 IETF - includes the year
SIEM & parsing of logs
LogQL & PromQL - are security holes these often overlooked
Anti-patterns for log management
Practical Application
- Only Katka is popular for MQ apps
ISO 8601 - timestamp standard
Journal ctl - not readable by human
Tools ; SC
SCAP
SIEM
DLP
SNMP - used to read data never to configure only version 3 used (2 or)

# UNIT 7

Monitoring & Alerting
TSDB - Time series database
Log data :
Firewalls, Applications, Endpoint, OS security, Network
Used for research of past or to predict future
Data sources
- Vulnerability scans, Automated reports, Dashboards,

LSG

# UNIT 8

Configuration & Drift management
Sometimes you fix the system
Sometimes you fix the users - retraining, help users avoid probs
As soon as we build system & roll them out ... the config starts to drift

Oracle systems require grid & Oracle user ids

1.3 Change management
ServiceNow (Snow)
If you build an allow list then you need a deny list
CMDB - Configuration Management Database
Documentation - keep updating

5.5 Audits & assessments
PCI DSS
White box PenTest
Gray box PenTest
Black box PenTest
Penetrating Test - Not fun (90% reports)

# UNIT 9

Certificate & keys
TLS - transport Layer Security

1.4 Cryptographic solutions
Key escrow - they keep the keys incase the CA goes out of business
Encrypt - tunnels

What is Identity
- a digital representation of a person
X.509
What is encryption
Symmetric - One way encryption, Same key on both sides
Assymetric - Two way encryption but One way algorithm
When do we encrypt
Envelope Encryption
AES 256

TLS 1.2 → 7 steps
TLS 1.3 → 5 steps

MAC Message Authentication Code

1.4 cont'd Obfuscation
- Steganography
- Tokenization
- Data masking (like hiding the first 5 #'s numbers in Social Security #)

5.3 Agreement types
SLA, MOA, MOU, MSA, SOW

5.4 Due diligence