Surveillances et Logs

Introduction :

Dans le cadre de la politique de sécurité informatique de **SmartTech**, la **surveillance active** et la **journalisation centralisée** des services critiques représentent un **pilier essentiel** pour garantir l'intégrité, la traçabilité et la disponibilité du système d'information.

Afin d'optimiser la supervision de l'ensemble des composants de l'infrastructure, **tous les logs générés par les services critiques sont redirigés vers un serveur dédié à la centralisation des journaux**. Ce dispositif permet de **consolider les informations de sécurité en un point unique**, facilitant ainsi leur exploitation, leur analyse et leur archivage.

Les services concernés par cette collecte centralisée sont les suivants :

- **LDAP / Kerberos** : gestion centralisée des identités et de l'authentification des utilisateurs ;
- **Asterisk** : gestion des communications VoIP internes ;
- **FreeRADIUS** : authentification réseau (Wi-Fi, VPN) adossée à l'annuaire LDAP ;
- **Serveurs Proxy (DNS et HTTP(S))** : contrôle des accès internet sortants et filtrage ;
- **Serveur DNS** : résolution des noms de domaine internes et externes ;
- **Nextcloud** : hébergement collaboratif et partage sécurisé de fichiers ;
- **auditd** : surveillance des accès aux fichiers sensibles, notamment ceux liés aux services LDAP/Kerberos et Nextcloud.

Objectifs :

- **Centraliser la gestion des logs** pour une visibilité complète, un accès unifié et une meilleure corrélation des événements ;
- **Renforcer la sécurité du système d'information**, en détectant rapidement les anomalies, les tentatives d'intrusion ou les accès non autorisés ;
- **Garantir la disponibilité et la fiabilité des services**, en anticipant les défaillances techniques et en facilitant les opérations de maintenance.

Les sections suivantes détaillent les solutions mises en œuvre pour la collecte, l'analyse et la centralisation des logs, les méthodes d'audit employées ainsi que les stratégies de réponse ou d'alerte définies pour chaque service.

1. Rsyslog

Rsyslog est une version améliorée de syslog, offrant plus de fonctionnalités comme le filtrage avancé, le transport TCP, la journalisation en base de données, etc.

Configuration :

```
┌─[practice@parrot]─[~]
└──╼ $sudo apt install -y rsyslog
```

Cette commande installe le service Rsyslog sur notre système

```
┌─[practice@parrot]─[~]
└──╼ $systemctl enable rsyslog
```

Cette commande active le démarrage automatique de Rsyslog au boot du système

```
 GNU nano 7.2                    /etc/rsyslog.conf

module(load="imuxsock") # provides support for local system logging
module(load="imklog")   # provides kernel logging support
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
```

- Charge le module **imuxsock** pour la journalisation locale via les sockets Unix
- Charge le module **imklog** pour capturer les logs du noyau Linux

- Charge le module **imudp** et **imtcp** pour recevoir des logs via **UDP** et **TCP**
- Configure l'écoute sur le port 514 en UDP ET TCP (port standard pour syslog)

```
 GNU nano 7.2          /etc/rsyslog.d/20-remote-logs.conf
################################################################################
#### Template de rangement : /var/log/remote/<hostname>/<service>.log     ####
################################################################################
$template RemoteLogs,"/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"

################################################################################
#### Règle générale : tous les logs reçus par le réseau vont là-bas       ####
################################################################################
*.* ?RemoteLogs
& stop
```

Cette configuration permet de **stocker les logs reçus par le réseau** dans des fichiers organisés par machine et par service.

template RemoteLogs,"/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"

➤ Définit un **modèle de chemin** pour ranger les logs :

- **%HOSTNAME% :** le nom de la machine distante
- **%PROGRAMNAME% :** le service (ex. : slapd , nextcloud, etc.)
  les fichiers seront rangés dans **/var/log/remote/NOM_MACHINE/NOM_SERVICE.log**
- ***.* ?RemoteLogs**
  ➤ Cette règle dit que **tous les logs reçus par le réseau** (*.*) doivent être enregistrés selon le modèle **RemoteLogs**.
- **& stop**
  ➤ Indique à **rsyslog** d'**arrêter le traitement** des logs une fois qu'ils sont enregistrés ici (évite les doublons).

```
┌─[practice@parrot]─[~]
└──╼ $sudo mkdir -p /var/log/remote
```

Nous Créons le répertoire **remote** pour y stocker les logs.

```
┌─[✗]─[practice@parrot]─[~]
└──╼ $sudo chmod 755 /var/log/remote
┌─[practice@parrot]─[~]
└──╼ $sudo systemctl restart rsyslog
```

Nous définissons les permission et redémarrons le service.

```
┌─[practice@parrot]─[~]
└──╼ $sudo netstat -tulnp | grep rsyslog
[sudo] password for practice:
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN
893/rsyslogd
tcp6       0      0 :::514                  :::*                    LISTEN
893/rsyslogd
udp        0      0 0.0.0.0:514             0.0.0.0:*
893/rsyslogd
udp6       0      0 :::514                  :::*
893/rsyslogd
```

rsyslog est désormais prêt pour recevoir les Logs des services.

1.1 Logs du Serveur DNS :

Nous installons rsyslog dans notre serveur dns.

```
phone1@phone:~$ sudo apt install -y rsyslog
```

```
phone1@phone:~$ sudo nano /etc/bind/named.conf.options
```

Nous éditons le fichier **named.conf.options** ajouter la commande ci-dessous.

```
logging {
    channel syslog_bind {
        syslog daemon;              // Facility "daemon"
        severity info;              // Niveau minimum à logger
    };

    category default { syslog_bind; };
    category queries { syslog_bind; };
    category resolver { syslog_bind; };
};
```

- Définit un canal de **logging** vers **syslog** avec:
  - Facility: **daemon** (catégorie système pour les démons)
  - Niveau: **info** (capture info, warning, error, etc.)
- Applique ce canal à:
  - Logs par défaut (**default**)
  - Requêtes DNS (**queries**)
  - Résolution DNS (**resolver**)

```
  GNU nano 7.2                  /etc/rsyslog.d/50-forward-logs.conf
# Envoi uniquement des logs liés à BIND (facility daemon) vers le serveur central
if $programname == 'named' then @@192.168.200.10:514
& stop
```

- Filtre les logs du programme named (BIND9)
- @@ signifie envoi via TCP (plus fiable que UDP avec @)
- Envoie vers le serveur 192.168.200.10 port 514
- & stop arrête le traitement ultérieur de ces logs

```
phone1@phone:~$ sudo systemctl restart bind9
sudo systemctl restart rsyslog
```

- Applique les modifications en redémarrant:
  - BIND9 (service DNS)
  - Rsyslog (service de gestion des logs)

```
phone1@Serveur-DNS-PRIMAIRE:~$ dig @127.0.01 voici_un_test_.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> @127.0.01 voici_un_test_.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 988
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 314a10d735bb3ce101000000685be588cf2aff91f4a6e141 (good)
; EDE: 18 (Prohibited)
;; QUESTION SECTION:
;voici_un_test_.com.            IN      A
```

Envoie d'un message de test pour verifier le bon fonctionnement de rsyslog:

Ce message devrait être capturé et envoyé au serveur central comme les logs BIND9

```
┌─[practice@parrot]─[/var/log/remote]
└──╼ $ls /var/log/remote/
parrot  phone  Serveur-DNS-PRIMAIRE
```

Le Dossier **Serveur-DNS-PRIMAIRE** a été automatiquement crée et c'est le **hostname** de notre Serveur DNS.

```
┌─[practice@parrot]─[/var/log/remote/Serveur-DNS-PRIMAIRE]
└──╼ $sudo grep -E "voici_un_test_.com" /var/log/remote/Serveur-DNS-PRIMAIRE/nam
ed.log
2025-06-25T08:03:20+00:00 Serveur-DNS-PRIMAIRE named[1714]: client @0x7fc5ae7611
68 127.0.0.1#51948 (voici_un_test_.com): query: voici_un_test_.com IN A +E(0)K (
127.0.0.1)
2025-06-25T08:03:20+00:00 Serveur-DNS-PRIMAIRE named[1714]: client @0x7fc5ae7611
68 127.0.0.1#51948 (voici_un_test_.com): query (cache) 'voici_un_test_.com/A/IN'
 denied (allow-query-cache did not match)
2025-06-25T08:03:20+00:00 Serveur-DNS-PRIMAIRE named[1714]: client @0x7fc5ae7611
68 127.0.0.1#51948 (voici_un_test_.com): query failed (REFUSED) for voici_un_tes
t_.com/IN/A at query.c:5688
```

Comme on peut le voir les logs ont bien été envoyer à notre serveur **rsyslog**.

1.2 Logs LDAP et Kerberos :

Nous installons rsyslog dans la machine.

```
phone1@kdc:~$ sudo apt install -y rsyslog
```

```
 GNU nano 7.2                    enable-logging.ldif
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: stats sync config
```

- **Explication** : Fichier LDIF (LDAP Data Interchange Format) pour modifier la configuration **OpenLDAP**
- **Détails** :
  - **dn: cn=config** : Cible la configuration globale
  - **changetype**: **modify** : Spécifie une modification
  - **replace**: **olcloglevel** : Remplace le paramètre de niveau de log
  - **olcloglevel**: **stats sync config** : Active les logs pour les statistiques, synchronisation et configuration

```
phone1@kdc:~$ sudo slapcat -n 0 | grep olcLogLevel
olcLogLevel: none
phone1@kdc:~$ sudo nano enable-logging.ldif
phone1@kdc:~$ ldapmodify -x -D "cn=admin,cn=config" -w "Passer123" -H ldap://loc
alhost -f enable-logging.ldif
modifying entry "cn=config"
```

Nous chargeons le fichier dans notre base avec la commande modify.

```
 GNU nano 7.2          /etc/rsyslog.d/50-forward-logs.conf
if $programname == 'slapd' then @@192.168.200.10:514
& stop
```

Dans le fichier **/etc/rsyslog.d/50-forward-logs.conf** nous spécifions que tout les logs de **slapd** (LDAP)

Doivent être envoyer a notre serveur des logs.

```
phone1@kdc:~$ hostnamectl
 Static hostname: kdc.tp.local
```

Nous voyons que le nom de la machine c'est kdc donc rsyslog devrait créer un repertoire portant ce nom.

```
┌─[practice@parrot]─[/var/log/remote]
└──╼ $ls
kdc  parrot  phone  Serveur-DNS-PRIMAIRE
```

Et effectivement c'est le cas.

```
┌─[✗]─[practice@parrot]─[~]
└──╼ $sudo tail -f /var/log/remote/kdc/slapd.log
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1003 op=1 BIND dn="uid=freeradius
_admin,ou=freeradius_users,dc=tp,dc=local" mech=SIMPLE bind_ssf=0 ssf=256
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1003 op=1 RESULT tag=97 err=0 qti
me=0.000020 etime=0.000132 text=
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1004 fd=20 ACCEPT from IP=192.168
.200.10:57800 (IP=0.0.0.0:389)
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1004 op=0 EXT oid=1.3.6.1.4.1.146
6.20037
```

Les Logs ont bien été enregistrer.

1.3 Logs Asterisk :

Nous installons rsyslog dans notre machine asterisk.

```
phone1@phone:~$ sudo apt install -y rsyslog
```

```
 GNU nano 7.2                  /etc/asterisk/logger.conf
; "logger reload" at the CLI will reload configuration
; of the logging system.

[general]
;
; Customize the display of debug message time stamps
; this example is the ISO 8601 date format (yyyy-mm-dd HH:MM:SS)
;
; see strftime(3) Linux manual for format specifiers.  Note that there is also
; a fractional second parameter which may be used in this field.  Use %1q
; for tenths, %2q for hundredths, etc.
;
dateformat=%F %T       ; ISO 8601 date format
```

Nous editons le fichier **/etc/asterisk/logger.conf**

- Standardise le format des timestamps pour une meilleure lisibilité
- Compatible avec les outils d'analyse (ELK, Splunk)
- Référence : **strftime(3)** pour personnalisation avancée

```
 GNU nano 7.2                  /etc/asterisk/logger.conf
;
[logfiles]
syslog.local0 => notice,warning,error,verbose
;
```

Fonctionnement :

- Redirige les logs **Asterisk** vers :
    - **Facility** : **local0** (dédié aux applications personnalisées)
    - **Niveaux** :
        - **notice** : Événements importants
        - **warning** : Avertissements
        - **error** : Erreurs critiques
        - **verbose** : Détails de débogage

```
 GNU nano 7.2          /etc/rsyslog.d/60-forward-asterisk.conf
local0.* @@192.168.200.10:514
```

Dans le fichier **60-forward-asterisk.conf** nous spécifions la destination des Logs.

```
phone1@phone:~$ hostnamectl
 Static hostname: Serveur-ToIP-Asterisk
```

```
phone1@phone:~$ sudo systemctl restart rsyslog
sudo systemctl restart asterisk
```

```
┌─[practice@parrot]─[/var/log/remote]
└──╼ $ls
kdc  parrot  phone  Serveur-DNS-PRIMAIRE  Serveur-ToIP-Asterisk
```

Le répertoire a effectivement été créer.

```
┌─[✗]─[practice@parrot]─[~]
└──╼ $sudo tail -f /var/log/remote/kdc/slapd.log
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1003 op=1 BIND dn="uid=freeradius
_admin,ou=freeradius_users,dc=tp,dc=local" mech=SIMPLE bind_ssf=0 ssf=256
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1003 op=1 RESULT tag=97 err=0 qti
me=0.000020 etime=0.000132 text=
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1004 fd=20 ACCEPT from IP=192.168
.200.10:57800 (IP=0.0.0.0:389)
2025-06-25T17:12:36+00:00 kdc slapd[940]: conn=1004 op=0 EXT oid=1.3.6.1.4.1.146
6.20037
```

1.4 Logs Proxy DNS :

Nous suivons le même processus pour **dnsmasq.**

```
/ # apt install -y rsyslog
```

```
/ # vi /etc/rsyslog.d/70-dnsmasq-file.conf
```

```
module(load="imfile")

input(type="imfile"
      File="/var/log/dnsmasq.log"
      Tag="dnsmasq"
      Severity="info"
      Facility="local2")

local2.* @@192.168.200.10:514module(load="imfile")

~
~
```

```
/ # rsyslogd
```

```
[practice@parrot]-[/var/log/remote]
    $ls
kdc         Nextcloud  phone                 Serveur-ToIP-Asterisk
Les-Proxy   parrot     Serveur-DNS-PRIMAIRE
```

```
[practice@parrot]-[/var/log/remote]
    $cd Les-Proxy/
[practice@parrot]-[/var/log/remote/Les-Proxy]
    $ls
dnsmasq.log
```

Nous faisons de meme pour le proxy http et freeradius

2. Auditd:

Auditd (audit daemon) est le démon du framework d'audit de sécurité intégré au noyau Linux. Il permet d'enregistrer de manière détaillée les activités système pour la surveillance, la conformité et le dépannage.

2.1 Configuation et test :

```
phone1@kdc:~$ sudo apt install auditd audispd-plugins
```

- **auditd** : Le démon principal d'audit pour Linux.
- **audispd-plugins** : Modules optionnels pour le traitement avancé des logs

```
phone1@kdc:~$ sudo systemctl enable auditd
Synchronizing state of auditd.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable auditd
phone1@kdc:~$ sudo systemctl start auditd
```

Nous activons **auditd** au démarrage avec la commande **systemctl enable auditd** puis activons le service avec **systemctl start auditd**.

```
 GNU nano 7.2                /etc/audit/rules.d/ldap.rules *
# Surveillance des accès au fichier slapd(ldap) et Kerb
-w /etc/ldap/ -p wa -k ldap_config
-w /var/lib/ldap/ -p wa -k ldap_data
-w /etc/krb5.conf -p wa -k kerberos_conf
-w /etc/krb5.keytab -p r -k kerberos_keytab
```

Cette configuration a pour but de **surveiller les activités sensibles** liées à :

- **LDAP** (protocole d'annuaire pour l'authentification)
- **Kerberos** (système d'authentification réseau)

Elle permet de détecter :

- Les modifications non autorisées des fichiers de configuration
- Les accès illégitimes aux données sensibles
- Les tentatives de compromission des services d'identité

```
phone1@kdc:~$ sudo augenrules --load
```

Cette commande Compile toutes les règles dans **/etc/audit/rules.d/.**

```
phone1@kdc:~$ sudo nano /etc/krb5.conf
phone1@kdc:~$ sudo ausearch -k kerberos_conf
----
time->Mon Jun 23 18:00:24 2025
type=PROCTITLE msg=audit(1750716024.928:138): proctitle=2F7362696E2F617564697463
746C002D52002F6574632F61756469742F61756469742E72756C6573
type=PATH msg=audit(1750716024.928:138): item=0 name="/etc/" inode=129793 dev=08
:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0 cap_f
e=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1750716024.928:138): cwd="/home/phone1"
type=SOCKADDR msg=audit(1750716024.928:138): saddr=100000000000000000000000
type=SYSCALL msg=audit(1750716024.928:138): arch=c000003e syscall=44 success=yes
 exit=1084 a0=3 a1=7ffe87114890 a2=43c a3=0 items=1 ppid=3084 pid=3098 auid=1000
 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=2 comm="au
ditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1750716024.928:138): auid=1000 ses=2 subj=unconfine
d op=add_rule key="kerberos_conf" list=4 res=1
```

Pour tester le bon fonctionnement de **kerberos** nous un des fichiers sensible (krb5.conf)

L'utilisateur phone1 a édité le fichier de configuration principal de Kerberos avec la commande : sudo **nano /etc/krb5.conf.**

La commande **ausearch** nous permet de savoir :

- **Qui a agi ?**
  - L'utilisateur **phone1** a utilisé **sudo** pour modifier un fichier sensible.
- **Quel fichier a été modifié ?**
  - Le fichier de configuration **/etc/krb5.conf** (configuration centrale de Kerberos).
- **Quand ?**
  - Le **23 juin 2025 à 18h00**.

2.2 Envoi des logs vers le serveur rsyslog pour la centralisation des logs :

```
 GNU nano 7.2              /etc/rsyslog.d/30-auditd.conf
module(load="imfile")  # une seule fois
input(type="imfile"
      File="/var/log/audit/audit.log"
      Tag="auditd"
      Severity="info"
      Facility="local6")
local6.* @@192.168.200.10:514
```

- Cette section indique à rsyslog de **surveiller le fichier /var/log/audit/audit.log**, qui est le journal des événements d'**auditd**.

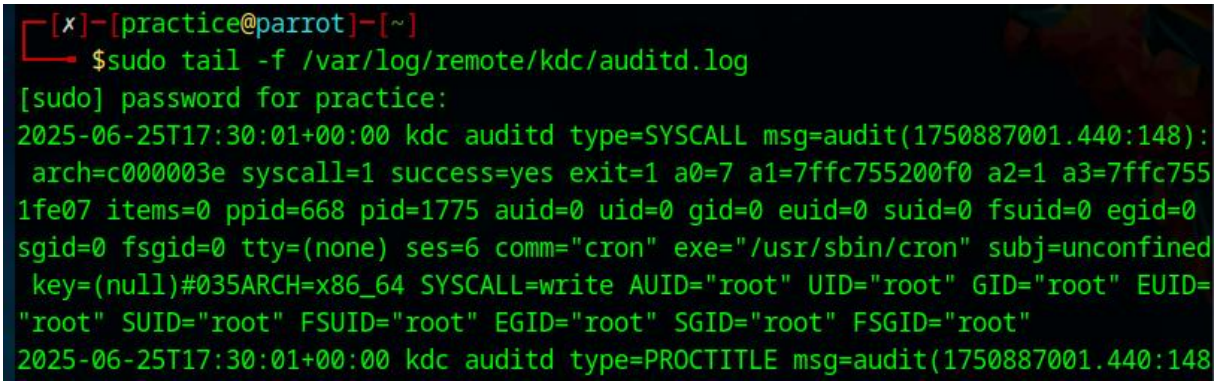
- **Tag="auditd"** : tous les messages lus auront ce tag, utile pour les identifier.

- **Severity="info"** : le niveau de gravité associé aux messages.

- **Facility="local6"** : la **"facility"** syslog utilisée pour classer les messages ; ici, local6 est une facility personnalisée (il en existe plusieurs : auth, daemon, local0 à local7, etc.).

Remarque : Une **facility** est un **champ de classification** des messages dans le protocole **Syslog**. Elle indique **l'origine ou la source du message de log**, comme par exemple le noyau, le système d'authentification, le courrier électronique, etc.

Les logs de **auditd** ont bien été envoyer vers notre serveur de logs depuis notre machine.



Nous faisons de même pour tout les autres service sensible comme Nextcloud , notre serveur de fichiers etc.

3. Conclusion :

La mise en place d'un système de **surveillance centralisée des journaux (logs) de nos 6 zones** représente une pierre angulaire dans la stratégie de sécurité informatique de SmartTech. En configurant correctement les services tels qu'**auditd** et **rsyslog**, nous assurons une **traçabilité complète des événements** critiques sur le réseau et les systèmes.

Ce mécanisme permet à l'entreprise :

- de **détecter rapidement les comportements anormaux** ou les tentatives d'intrusion,
- de **conserver des preuves** en cas d'incident de sécurité,
- de **se conformer aux exigences réglementaires** (ex. RGPD, ISO 27001),
- et de **faciliter le travail des administrateurs systèmes** grâce à une centralisation efficace des journaux