Körresultat för alla skript

1. `log_analysis.py` (Python)

Syfte

Analysera systemloggar och upptäcka misstänkta inloggningsförsök.

Exempel på körresultat (`security_report.txt`)

```
2026-01-16 09:12:01 - Log analysis started
2026-01-16 09:12:01 - Failed login attempts detected: 7
2026-01-16 09:12:01 - WARNING: Potential brute-force attack detected
2026-01-16 09:12:01 - Security log analysis completed
```

Alternativt felutfall

```
2026-01-16 09:10:44 - Log analysis started
2026-01-16 09:10:44 - ERROR: Log file system.log not found
```

2. `security_audit.ps1` (PowerShell)

Syfte

Grundläggande säkerhetsrevision av Windows-system
(t.ex. användarkonton, brandvägg, uppdateringar)

Exempel på konsolutskrift

```
[INFO] Starting Windows security audit...
[OK] Windows Firewall: Enabled
[OK] Automatic Updates: Enabled
[WARNING] Local Administrator account is enabled
[INFO] Number of local users: 6
[INFO] Security audit completed
```

Exempel på loggfil

```
2026-01-16 09:20:11 - Firewall status: Enabled
2026-01-16 09:20:11 - Automatic updates: Enabled
2026-01-16 09:20:11 - WARNING: Administrator account enabled
```

3. `security_check.sh` (Bash)

Syfte

Snabb säkerhetskontroll av Linux-system
(tjänster, rättigheter, SSH-konfiguration)

Exempel på terminalutskrift

```
[INFO] Starting Linux security check
[OK] SSH root login: disabled
[OK] Firewall (ufw): active
[WARNING] World-writable files found: /tmp/test.log
[INFO] Running services: 37
[INFO] Security check completed
```

Exempel på sammanfattande status

```
RESULT: WARNINGS DETECTED
- World-writable files present
```

Sammanfattning i tabell

Skript	Plattform	Resultattyp	Säkerhetsvarning
`log_analysis.py`	Linux / generisk	Loggfil	Ja
`security_audit.ps1`	Windows	Konsol + logg	Ja
`security_check.sh`	Linux	Konsol	Ja

Teknisk dokumentation

Säkerhetskontroller och logganalys via automatiserade skript

1. Översikt

Denna dokumentation beskriver tre automatiserade säkerhetsskript avsedda för:

- logganalys och detektion av misstänkt aktivitet
- grundläggande säkerhetskontroller på Linux- och Windows-system
- stöd för incidentidentifiering och förebyggande arbete

Skripten är avsedda att köras fristående eller integreras i automatiserade drift- och säkerhetsflöden.

2. Ingående komponenter

Skript	Plattform	Språk	Huvudfunktion
`log_analysis.py`	Plattformoberoende	Python	Analys av autentiseringsloggar
`security_audit.ps1`	Windows	PowerShell	Säkerhetsrevision av systeminställningar
`security_check.sh`	Linux	Bash	Snabb säkerhetskontroll av system

3. `log_analysis.py` – Logganalys och anomalidetektion

3.1 Syfte

Identifiera potentiella brute-force-attacker genom att analysera systemloggar och räkna misslyckade inloggningsförsök.

3.2 Funktionell beskrivning

- Läser angiven loggfil (`system.log`)
- Identifierar rader som innehåller:
    - `"Failed password"`
    - `"failed login"`
- Räknar antal förekomster
- Jämför mot ett definierat tröskelvärde
- Skriver resultat och varningar till loggfil

3.3 Konfiguration

```
LOGFILE = "security_report.txt"
FAILED_THRESHOLD = 5
```

3.4 Utdata

- Fil: `security_report.txt`
- Innehåll:
    - tidsstämplade statusmeddelanden
    - antal misslyckade inloggningar
    - varning vid överskridet tröskelvärde

3.5 Felhantering

- Om loggfil saknas loggas fel
- Skriptet avslutas kontrollerat utan krasch

4. `security_audit.ps1` – Windows säkerhetsrevision

4.1 Syfte

Utföra grundläggande säkerhetskontroller på Windows-system för att identifiera vanliga svagheter.

4.2 Typiska kontroller

- Status för Windows-brandvägg
- Automatiska uppdateringar
- Lokala användarkonton
- Administratörskonton
- Grundläggande systemkonfiguration

4.3 Utdata

- Konsolutskrift med statusmeddelanden (`[OK]`, `[WARNING]`, `[INFO]`)
- Valfri loggfil för revisionsspårbarhet

4.4 Användningsområde

- Manuell säkerhetskontroll
- Driftkontroller
- Underlag för intern revision

5. `security_check.sh` – Linux säkerhetskontroll

5.1 Syfte

Genomföra en snabb översiktlig säkerhetskontroll av ett Linux-system.

5.2 Typiska kontroller

- SSH-konfiguration (root-inloggning)
- Brandväggsstatus
- Filrättigheter (world-writable filer)
- Aktiva tjänster och processer

5.3 Utdata

- Terminalutskrift med tydliga statusmeddelanden
- Sammanfattande resultat (OK / WARNING)

5.4 Förutsättningar

- Bash-shell
- Grundläggande systemverktyg (`grep`, `awk`, `find`, `systemctl`)

6. Säkerhetsklassning av resultat

Resultat	Betydelse
OK	Ingen åtgärd krävs
INFO	Informativt värde
WARNING	Rekommenderad åtgärd
ERROR	Fel som bör åtgärdas omgående

7. Begränsningar

- Skripten utför **indikativ analys**, inte fullständig forensik
- Tröskelvärden är statiska

- Ingen central korrelation mellan system
- Kräver manuellt eller externt orkestrerad schemaläggning

8. Rekommenderad vidareutveckling

- Central logginsamling
- Tidsbaserad trendanalys
- Exit codes för CI/CD-integration
- JSON-utdata för SIEM/SOAR
- Automatisk rapportgenerering

9. Sammanfattning

Skripten utgör ett lättviktigt men effektivt ramverk för:

- grundläggande säkerhetsövervakning
- tidig varningsdetektion
- stöd till drift- och säkerhetsarbete

Dokumentationen är avsedd att kunna användas direkt i tekniska miljöer utan ytterligare anpassning.