

# Projet de recherche

Automatisation de la détection de  
latéralisation avec Kestrel

Vincent Cardile  
Léon Gall  
Hugo Himber  
Quentin Nagel

# Contexte

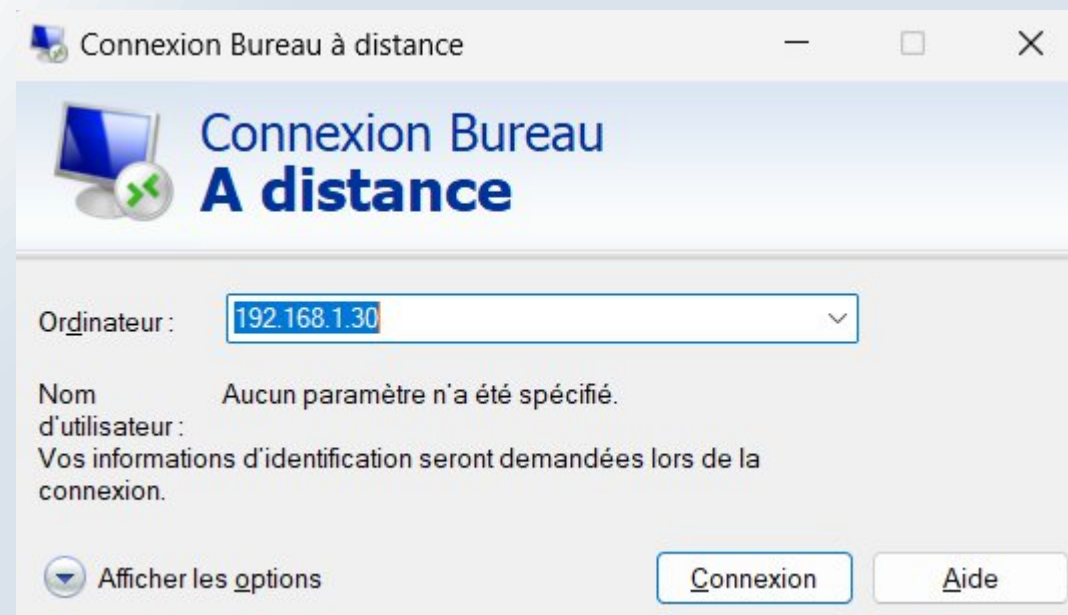
- Latéralisation
  - Mouvement d'un attaquant dans un réseau
  - Objectifs :
    - Reconnaissance des actifs et données accessibles
    - Élévation de privilèges
    - Persistance de l'attaque
  - Exploitation de divers protocoles comme
    - RDP, SSH, SMB
- Exemple d'attaque
  - WannaCry (2017)
    - Énumération des sessions RDP pour se propager

# Remote Desktop Protocol (RDP)

- Protocole de contrôle de poste à distance
- Environnement graphique complet
- Développé par Microsoft pour Windows
- Port TCP 3389

## Pourquoi RDP ?

- SSH désactivé par défaut
- Prévalent en entreprise
- Ciblé par les attaquants



# Problématiques

- Détection complexe selon le type d'attaque
  - Détournement de sessions et exploitation de vulnérabilités
    - Présence d'Indicateur de compromission (IoC)
  - Attaque par identifiant compromis
    - Analyse de comportements
- Grande quantité de données
  - Plusieurs centaines d'événements par minute
  - Complexe à stocker et analyser rapidement

# Caldera & Kestrel



## Caldera

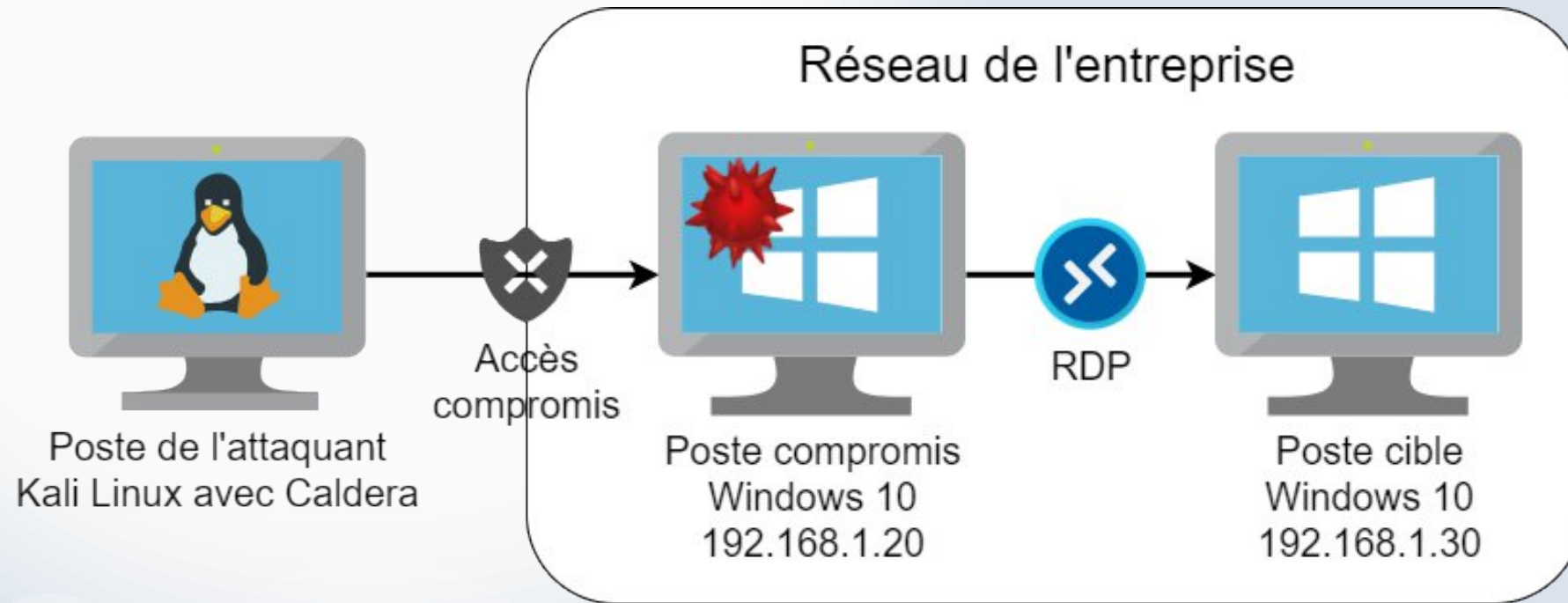
- Framework de simulations d'adversaires développé par Mitre
- Teste les capacités de détection et de réponses aux attaques
- Joue le rôle de commande et contrôle (C2)



## Kestrel

- Langage de chasse aux menaces initié par IBM
- Libre de droit
- Interfaçable avec d'autre technologie pour la détection

# Architecture de l'attaque



## Préparation de l'attaque :

Accès initial au réseau + identifiants seconde victime

# Déroulement de l'attaque

## Exemple de scénario :

1. Reconnaissance : Prise d'informations sur la 1ère victime
2. Accès initial : Obtention d'un accès sur le poste de la première victime et installation de l'agent Caldera
3. Mouvement latéral : Obtention des identifiants de la 2ème victime, puis connexion via RDP à son poste
4. Exécution : Lancement d'un script PowerShell pour installer l'agent Caldera sur le poste de la 2ème victime.
5. Exfiltration : Exfiltration de données ou latéralisation










# Déroulement de l'attaque

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Adversary-in-the-Middle	Account Discovery	Exploitation of Remote Services	Adversary-in-the-Middle	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Gather Victim Host Information	Acquire Infrastructure	Drive-by Compromise	Command and Scripting Interpreter	BITS Jobs	Access Token Manipulation	Access Token Manipulation	Brute Force	Application Window Discovery	Internal Spearfishing	Archive Collected Data	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information	Compromise Accounts	Exploit Public-Facing Application	AppleScript	Boot or Logon Autostart Execution	Account Manipulation	BITS Jobs	Credentials from Password Stores	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltration Over Alternative Protocol	Data Encrypted for Impact
Gather Victim Network Information	Compromise Infrastructure	External Remote Services	AutoHotKey & AutoIT	Boot or Logon Initialization Scripts	Boot or Logon Autostart Execution	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking	Automated Collection	Data Encoding	Exfiltration Over C2 Channel	Data Manipulation
Gather Victim Org Information	Develop Capabilities	Hardware Additions	Cloud API	Browser Extensions	Boot or Logon Initialization Scripts	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services	Browser Session Hijacking	Data Obfuscation	Exfiltration Over Other Network Medium	Defacement
Phishing for Information	Establish Accounts	Phishing	JavaScript	Compromise Host Software Binary	Create or Modify System Process	Decfuscate/Decode Files or Information	Forge Web Credentials	Cloud Service Discovery	Cloud Services	Clipboard Data	Dynamic Resolution	Exfiltration Over Physical Medium	Disk Wipe
Search Closed Sources	Obtain Capabilities	Spearfishing Attachment	Lua	Create Account	Domain or Tenant Policy Modification	Deploy Container	Input Capture	Cloud Storage Object Discovery	Direct Cloud VM Connections	Data from Cloud Storage	Encrypted Channel	Exfiltration Over Web Service	Endpoint Denial of Service
Search Open Technical Databases	Stage Capabilities	Spearfishing Link	Network Device CLI	Create or Modify System Process	Escape to Host	Direct Volume Access	Modify Authentication Process	Container and Resource Discovery	Distributed Component Object Model	Data from Configuration Repository	Fallback Channels	Scheduled Transfer	Financial Theft
Search Open Websites/Domains		Spearfishing via Service	PowerShell	Event Triggered Execution	Event Triggered Policy Modification	Domain or Tenant Policy Modification	Multi-Factor Authentication Interception	Debugger Evasion	Remote Desktop Protocol	Data from Information Repositories	Hide Infrastructure	Transfer Data to Cloud Account	Firmware Corruption
Search Victim-Owned Websites		Spearfishing Voice	Python	External Remote Services	Exploitation for Privilege Escalation	Execution Guardrails	Multi-Factor Authentication Request Generation	Device Driver Discovery	SMB/Windows Admin Shares	Data from Local System	Ingress Tool Transfer		Inhibit System Recovery
		Replication Through Removable Media	Unix Shell	Hijack Execution Flow	Hijack Execution Flow	Exploitation for Defense Evasion	Network Sniffing	Domain Trust Discovery	SSH	Data from Network Shared Drive	Multi-Stage Channels		Network Denial of Service
		Supply Chain Compromise	Visual Basic	Implant Internal Image	Process Injection	File and Directory Permissions Modification	OS Credential Dumping	File and Directory Discovery	VNC	Data from Removable Media	Non-Application Layer Protocol		Resource Hijacking
		Trusted Relationship	Windows Command Shell	Modify Authentication Process	Scheduled Task/Job	Hide Artifacts	Steal Application Access Token	Group Policy Discovery	Windows Remote Management	Data Staged	Non-Standard Port		Service Stop
		Valid Accounts	Container Administration Command	Office Application Startup	Valid Accounts	Hijack Execution Flow	Steal or Forge Authentication Certificates	Log Enumeration	Replication Through Removable Media	Email Collection	Protocol Tunneling		System Shutdown/Reboot
			Deploy Container	Power Settings		Impair Defenses	Steal or Forge Kerberos Tickets	Network Service Discovery	Software Deployment Tools	Input Capture	Proxy		
			Exploitation for Client Execution	Pre-OS Boot		Impersonation	Steal Web Session Cookie	Network Share Discovery	Taint Shared Content	Screen Capture	Remote Access Software		
			Inter-Process Communication	Scheduled Task/Job		Indicator Removal	Unsecured Credentials	Network Sniffing	Use Alternate Authentication Material	Video Capture	Traffic Signaling		
			Native API	Server Software Component		Indirect Command Execution		Password Policy Discovery			Web Service		
			Scheduled Task/Job	Traffic Signaling		Masquerading		Peripheral Device Discovery					



# Résultats de l'attaque

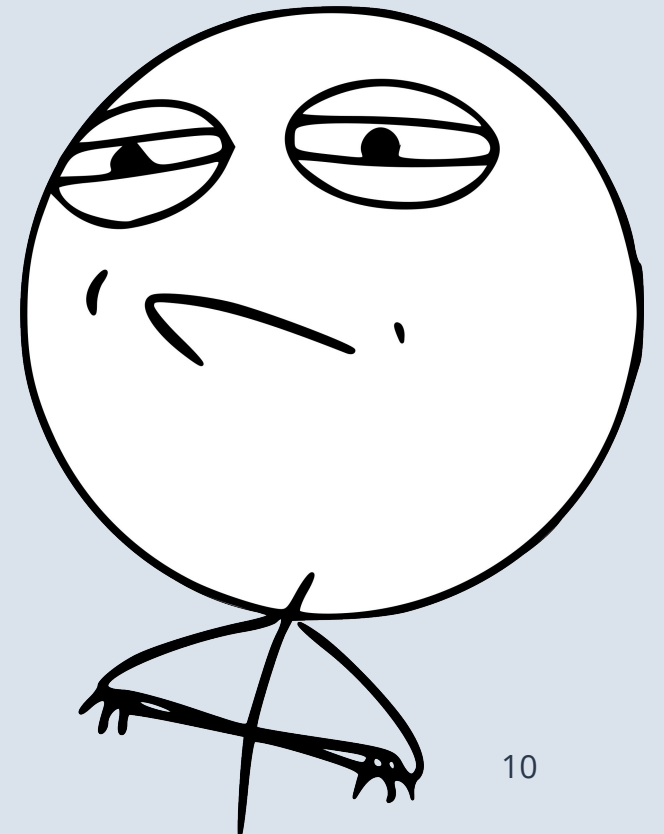
- Ajout d'un nouvel agent sur l'interface Caldera
- Ajout d'un fichier *coucou.txt* sur le poste cible
- Ajout de journaux à exploiter pour détecter l'attaque

Operational				
Nombre d'événements : 4 238 (!) Nouveaux événements disponibles				
Niveau	Date et heure	Source	ID de l'événement	Catégorie de la tâche
 Information	03/04/2025 06:23:25	Sysmon	26	File Delete logged (rule: FileDeleteDetected)
 Information	03/04/2025 06:23:25	Sysmon	26	File Delete logged (rule: FileDeleteDetected)
 Information	03/04/2025 06:23:16	Sysmon	11	File created (rule: FileCreate)
 Information	03/04/2025 06:23:15	Sysmon	1	Process Create (rule: ProcessCreate)
 Information	03/04/2025 06:23:15	Sysmon	1	Process Create (rule: ProcessCreate)
 Information	03/04/2025 06:23:15	Sysmon	26	File Delete logged (rule: FileDeleteDetected)
 Information	03/04/2025 06:23:15	Sysmon	1	Process Create (rule: ProcessCreate)

id (paw)	host	group	platform	contact	pid	privilege	status	last seen
pyafoc	victim1	red	windows	HTTP	7876	Elevated	alive, trusted	3/29/2025, 6:45:04 PM
avcjl	victim2	red	windows	HTTP	5940	User	alive, trusted	3/29/2025, 6:44:52 PM

# Détection : quelques événements pertinents

- Connexion réussie : 4624, Type 10 (Distant)
- Session reconnectée : 4778 , Type 10
- Session déconnectée : 4779 , Type 10
- Création de processus : 4688 et argument « CommandLine » Sysmon
- Création de fichier : Sysmon 11



# Détection : Mise en place

Les requêtes Kestrel

- Similaires au SQL :
  - Une source de données
  - Une clé
  - Les identifiants de sélection

Source de données

- Journaux evtx windows
- Nécessite de les convertir en CSV

```
rdp_events = get windows_event  
              from win  
              where EventID = "4624"  
              and LogonType = "10"
```

# Détection : Exemples

```
test_ip = get windows_event
          from rdp_events
          where SourceIP not in ["192.168.1.22", "192.168.1.21"]
```

- IP suspectes

baf23faa-a3d7-4cc1-b864-6c97c62d0742	52	2025-04-03 03:28:27.573469	4624	VICTIME2\$	10.0	192.168.1.20	VICTIME2	victim2
abf0c190-0ca7-40f2-9f4d-8259610f2cf1	53	2025-04-03 03:28:27.573486	4624	VICTIME2\$	10.0	192.168.1.20	VICTIME2	victim2

# Détection : Exemples

- Détection de *coucou.txt*

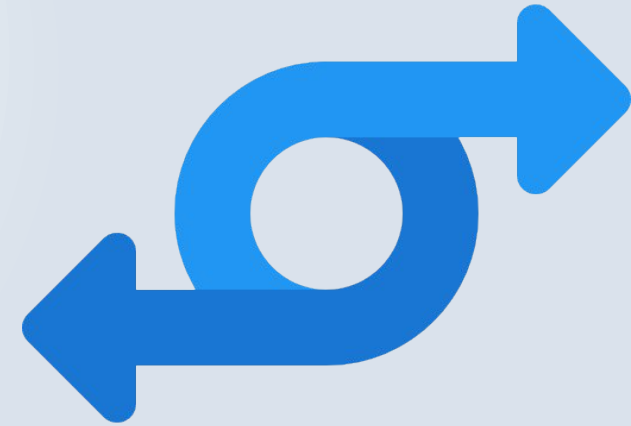
```
file_creation = get windows_event
                from sysmon
                where EventID = "11"
                and TargetFilename like "%coucou%"

disp file_creation attr EventID, UtcTime, Image, User, TargetFilename
```

EventID	UtcTime	Image	User	TargetFilename
11	2025-04-03 03:34:11.452	C: \Windows\Explorer.EXE	VICTIME2\victim2	C: \Users\victim2\Desktop\coucou.txt

# Détection : automatisation








- Notebook Jupiter
  - Processus prédéfinis
  - Chasse ciblée
- Exécution automatique
  - CI/CD
  - Cron Job
  - Alerte automatique



# Détection : automatisation POC

## GitLab CI/CD

- Kestrel utilisé via un conteneur
- Exécute un script Python grâce à conteneur docker personnalisé
- Génère un journal d'événements pertinents

Status	Pipeline	Created by	Stages
 Passed 🕒 00:00:21 📅 1 hour ago	upload rapport #361939  main  5c1330f3  latest		 



# Détection : automatisation POC

- Exemple

- Détection de connexion en heure creuse
- Effectue une alerte en cas de positif
- Génère un rapport d'événements pertinents

```
now = datetime.date.today()
today = datetime.datetime(now.year, now.month, now.day, 8, 0, 0)
"""Code à utiliser pour la chasse automatisé
target_time = now - datetime.timedelta(days=1)
yesterday = datetime.datetime(target_time.year, target_time.month, target_time.day, 20, 0, 0)
"""
yesterday = datetime.datetime(2025,4,3,3,31,0,000000)

with Session() as session2:
    kestrel_query = f"""
events = LOAD "kestrel/win.csv" AS windows_event
rdp_events = get windows_event
                from events
                where EventID = "4624"
                and LogonType = "10"
                and TimeCreated < "{today}"
                and TimeCreated > "{yesterday}"

disp rdp_events
"""

    session2.execute(kestrel_query)
    rdp_events2 = session2.get_variable("rdp_events")
    if(rdp_events2):
        print("DANGER")
    else:
        print("Pas de danger")
```

# Détection : automatisation POC

Executing "step\_script" stage of the job script

00:02

```
$ echo "Exécution du script 'auto_detect.py'..."
```

```
Exécution du script 'auto_detect.py'...
```

```
$ python kestrel/auto_detect.py | tee kestrel/hunt_results.txt
```

DANGER

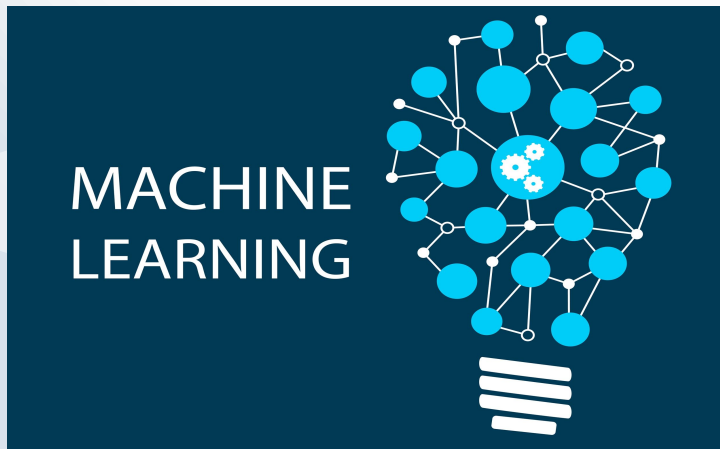
```
event 1 : {'id': '1b741c6d-75df-47e5-99e4-7aec8b9dea19', 'RecordNumber': 138, 'TimeCreated': '2025-04-03 03:31:18.685043',  
'EventID': 4624, 'AccountName': 'VICTIME2$', 'LogonType': 10.0, 'SourceIP': '192.168.1.20', 'DestinationIP': 'VICTIME2', 'TargetUserName': 'victime2', 'LogonId': None, 'ObjectName': None, 'AccessMask': None, 'FileName': None, 'FilePath': None, 'type': 'windows_event'}
```

```
event 2 : {'id': 'f473344c-097a-40a6-927a-4a6c7df112c9', 'RecordNumber': 139, 'TimeCreated': '2025-04-03 03:31:18.685062',  
'EventID': 4624, 'AccountName': 'VICTIME2$', 'LogonType': 10.0, 'SourceIP': '192.168.1.20', 'DestinationIP': 'VICTIME2', 'TargetUserName': 'victime2', 'LogonId': None, 'ObjectName': None, 'AccessMask': None, 'FileName': None, 'FilePath': None, 'type': 'windows_event'}
```

```
$ echo "test fichier d'enregistrement d'incident"
```

# Pour aller plus loin

- Intégration STIX/TAXII
  - Plus de conversion en CSV
  - Détection plus fine



- Apprentissage automatisé (ML)
  - Fiabilité
  - Performance matériel
  - Coût

# Conclusion

- Détection complexe
  - Quantité de données
  - Détection de comportements
- Efficacité de détection avec Kestrel
  - Détection basée sur des facteurs prédéterminés répétable
  - Performance temporelle dépendante de la quantité de journaux
- PoC d'automatisation de détection

**Merci de votre attention !  
Avez-vous des questions\* ?**

*\*Des conditions s'appliquent. Offre à durée limitée — un temps maximal de 5 minutes est alloué à la réponse aux questions.*



# Bibliographie et Références

- **Dutta, T.S.** (2025). Hackers used weaponized zoom installer to gain RDP access & deploy BlackSuit ransomware. *Cyber Security News*. <https://cybersecuritynews.com/hackers-used-weaponized-zoom-installer/>
- **The DFIR Report.** (2025). Fake zoom ends in BlackSuit ransomware. <https://thedfirreport.com/2025/03/31/fake-zoom-ends-in-blacksuit-ransomware/>
- **Ozarslan, S.** (2025). Tactics, techniques, and procedures (TTPs) used in the SolarWinds breach. *Picus Security Blog*. <https://www.picussecurity.com/resource/blog/ttps-used-in-the-solarwinds-breach>
- **Palo Alto Networks.** (2025). What is lateral movement?. <https://www.paloaltonetworks.ca/cyberpedia/what-is-lateral-movement>
- **Cloudflare.** (2025). What is lateral movement in cyber security?. <https://www.cloudflare.com/learning/security/glossary/what-is-lateral-movement/>
- **Baker.** (2025). What is lateral movement?. *CrowdStrike*. <https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/lateral-movement/>
- **Mabrouk, A.** (2024). Lateral movement attacks datasets: benchmarking, challenges, and solutions. *MSc Thesis, University of Windsor*. (URL non fournie dans la source originale)
- **Microsoft Learn.** (2025). Présentation du protocole RDP (Remote Desktop Protocol). <https://learn.microsoft.com/fr-fr/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol>
- **MITRE ATT&CK.** (2025). MITRE ATT&CK Framework. <https://attack.mitre.org/>
- **Smiliotopoulos, C., Kambourakis, G., & Kolias, C.** (2024). Detecting lateral movement: A systematic survey. *Heliyon*, 10(4). DOI: 10.1016/j.heliyon.2024.e26317 (ou <https://doi.org/10.1016/j.heliyon.2024.e26317>)
- **Kambourakis, G., Kolias, C., & Stavrou, A.** (2017). The Mirai botnet and the IoT Zombie Armies. *MILCOM 2017*. <https://ieeexplore.ieee.org/document/8170867> (DOI: 10.1109/MILCOM.2017.8170867)
- **Smiliotopoulos, C., & Kambourakis, G.** (2023). "LMD" sysmon dataset collections. *GitHub Repository*. [https://github.com/ChristosSmiliotopoulos/Lateral-Movement-Dataset--LMD\\_Collections](https://github.com/ChristosSmiliotopoulos/Lateral-Movement-Dataset--LMD_Collections)
- **Smiliotopoulos, C., Barmapsalou, K., & Kambourakis, G.** (2022). Revisiting the detection of lateral movement through sysmon. *Applied Sciences*, 12(15). <https://www.mdpi.com/2076-3417/12/15/7746> (DOI: 10.3390/app12157746)
- **Bai, T., Bian, H., Salahuddin, M. A., Abou Daya, A., Limam, N., & Boutaba, R.** (2021). RDP-based Lateral Movement detection using Machine Learning. *Computer Communications*, 165. DOI: 10.1016/j.comcom.2020.10.013 (ou <https://doi.org/10.1016/j.comcom.2020.10.013>)
- **Bai, T., Bian, H., Daya, A. A., Salahuddin, M. A., Limam, N., & Boutaba, R.** (2019). A Machine Learning Approach for RDP-based Lateral Movement Detection. *IEEE LCN 2019*. DOI: 10.1109/LCN44214.2019.8990853 (ou <https://doi.org/10.1109/LCN44214.2019.8990853>)
- **He, D., Gu, H., Zhu, S., Chan, S., & Guizani, M.** (2023). A Comprehensive Detection Method for the Lateral Movement Stage of APT Attacks. *IEEE IoT Journal*, 11(5). DOI: 10.1109/JIOT.2023.3322412 (ou <https://doi.org/10.1109/JIOT.2023.3322412>)
- **Cardile, V., Gall, L., Himber, H., & Nagel, Q.** (2025). INF808 - Latéralisation. *GitLab Repository*. <https://gitlab.unistra.fr/lgall/inf808-lateralisation>
- **Python Documentation.** (2025). email: Examples. <https://docs.python.org/3/library/email.examples.html>
- **Slack API / GitHub.** (2025). python-slack-sdk. *GitHub Repository*. <https://github.com/slackapi/python-slack-sdk>