CHAPTER 12

Lateral Movement Analysis

Chapter Overview

01

SMB & Pass-the-Hash

NTLM credential theft, Mimikatz, and PtH detection

02

Kerberos Attacks

Golden/Silver tickets, Kerberoasting, Pass-the-Ticket

03

Remote Execution Methods

PsExec, tasks, services, WMI, RDP, Remoting, SSH

SMB & Pass-the-Hash

NTLM credential reuse without cracking

SMB Protocol & NTLM Authentication

SMB (Server Message Block): Windows file/printer sharing — backbone of lateral movement

NTLM Challenge-Response: server sends 8-byte challenge; client responds with hash(password+challenge)

NTLM is used as fallback when Kerberos fails (wrong time, no DC reachable, IP address used)

Net-NTLMv1/v2: hashes captured on the wire (Responder) — crackable offline

NTLM hash (from SAM/LSASS): used directly for Pass-the-Hash — no cracking needed

LM hashes: legacy format, easily cracked — disable via GPO (NoLMHash policy)

Authentication flow: Type 1 (negotiate) → Type 2 (challenge) → Type 3 (authenticate)

Pass-the-Hash — Technique & Detection

Pass-the-Hash (PtH): present NTLM hash instead of password — Windows accepts it directly

Tools: Mimikatz pth, Impacket psexec.py, CrackMapExec, Evil-WinRM

Credential sources: LSASS memory dump, SAM hive, NTDS.dit (domain controller)

Mimikatz command: sekurlsa::logonpasswords — dumps all credentials from LSASS

Detection — Event 4624: Logon Type 3, Authentication Package = NTLM, unexpected source IP

Detection — Event 4648: Explicit credential use (run-as, PtH tools often trigger this)

Mitigation: Credential Guard, disable NTLM for domain auth, tiered admin model

LAPS (Local Admin Password Solution): randomize local admin passwords per machine

Credential Dumping Methods

From Memory (LSASS)

Mimikatz: sekurlsa::logonpasswords (requires debug privilege)

Task Manager: create minidump of lsass.exe → parse offline

ProcDump: procdump.exe -ma lsass.exe lsass.dmp

Detection: Event 4663 (object access on lsass), Sysmon Event 10

Credential Guard: moves LSASS into isolated VM — Mimikatz returns '*'

From Disk & Domain

SAM hive: local account hashes (system + SAM needed together)

reg save HKLM\SAM sam.hive + HKLM\SYSTEM system.hive

NTDS.dit: Active Directory database — all domain account hashes

DCSync (Mimikatz): lsadump::dcsync /domain /all — no file needed

Detection: Event 4662 (replication rights exercised by non-DC account)

Kerberos Attacks

Ticket theft, forgery, and service exploitation

Kerberos Authentication Flow

Step 1 — AS-REQ: client requests TGT from Key Distribution Center (KDC/DC)

Step 2 — AS-REP: KDC returns TGT encrypted with KRBTGT account hash

Step 3 — TGS-REQ: client presents TGT, requests service ticket (TGS) for target

Step 4 — TGS-REP: KDC returns TGS encrypted with target service account's hash

Step 5 — AP-REQ: client presents TGS to target service — access granted

Key insight: TGT is signed with KRBTGT — steal or forge it = own the domain

Key insight: TGS is signed with service account hash — steal or forge it = own the service

Pass-the-Ticket & Overpass-the-Hash

Pass-the-Ticket (PtT)

Steal TGT or TGS from memory using Mimikatz (sekurlsa::tickets)

Import ticket on attacker machine: kerberos::ptt ticket.kirbi

No password or hash needed — ticket IS the credential

TGT valid for 10 hours by default; TGS valid for service lifetime

Detection: Event 4768/4769 from unexpected source workstation

Overpass-the-Hash

Convert NTLM hash → Kerberos TGT (avoids NTLM-based detection)

Mimikatz: sekurlsa::pth /user:admin /ntlm:HASH /domain:corp

Opens cmd.exe with injected Kerberos context — requests legitimate TGT

All subsequent auth uses Kerberos — bypasses NTLM-only monitoring

Detection: 4768 (TGT request) from workstation not logged in as that user

Golden Ticket & Silver Ticket Attacks

Golden Ticket: forge a TGT using the KRBTGT account hash — grants any access in the domain

- **KRBTGT hash obtained by**: DCSync (lsadump::dcsync) or NTDS.dit extraction
- Forged TGT can have any username, groups, and a 10-year lifetime
- Survives all password resets EXCEPT resetting KRBTGT password TWICE (invalidates old tickets)

Detection: Event 4672 (special privileges) for accounts that shouldn't have them

Silver Ticket: forge a TGS for a specific service using the service account hash

- Silver Ticket requires no KDC contact — entirely offline, harder to detect

Scope: limited to one service vs. Golden Ticket's domain-wide access

Kerberoasting — In Depth

Concept: any authenticated domain user can request a TGS for any SPN in the domain

TGS is encrypted with the service account's RC4 (NTLM) hash by default

Attack: request ticket, extract from memory, crack offline with hashcat/john

Impacket: GetUserSPNs.py DOMAIN/user:pass -request — retrieves all crackable tickets

Rubeus: rubeus.exe kerberoast — runs in-memory, outputs ready-to-crack hashes

Detection: Event 4769, EncryptionType = 0x17 (RC4-HMAC) from a non-service account

Mitigation: enable AES-only Kerberos (msDS-SupportedEncryptionTypes = 24), use gMSA

Mitigation: audit SPNs quarterly — remove orphaned service accounts with SPNs

Remote Execution Methods

How attackers spread from host to host

PsExec, Scheduled Tasks & Services

PsExec

Copies executable to ADMIN$ share, creates PSEXESVC service, executes as SYSTEM

Detection: Event 7045 (service installed) with ServiceName = PSEXESVC

Detection: named pipe connection \\HOSTNAME\pipe\PSEXESVC in network logs

Mimikatz + PsExec: extract hash → PtH → PsExec → SYSTEM shell anywhere

Impacket psexec.py / smbexec.py: fileless variants, different artifact footprint

Scheduled Tasks & Services

schtasks /create /s HOST /tn name /tr command /sc once /st HH:MM

Detection: Event 4698 (task created) and 4702 (task modified) on remote host

sc \\HOST create svcname binpath= C:\mal.exe — installs service remotely

Detection: Event 7045 on target host, Event 4697 in Security log

Cleanup: attackers delete tasks/services after use — monitor 4699 (task deleted)

WMI, RDP, WinRM & SSH Pivots

WMI & RDP

WMI: wmic /node:HOST process call create 'cmd /c payload.exe'

Result: child of WmiPrvSE.exe — stealthy, no new service created

Detection: Event 4688 showing parent wmiprvse.exe spawning unusual children

RDP: Type 10 logon — credentials cached on target unless Restricted Admin Mode

Detection: Event 4778 (reconnect) and 4779 (disconnect) on target

WinRM / PSRemoting & SSH

Invoke-Command / Enter-PSSession: Type 3 logon — credentials NOT cached

Detection: Event 4688 showing wsmprovhost.exe as parent on target

PowerShell script block logging (4104) captures all commands run remotely

SSH tunnels: ssh -L localport:internal_target:port jump_host

Dynamic forwarding (SOCKS): ssh -D 1080 jump_host → entire subnet accessible

Detection: unusual SSH connections, non-standard source ports, long-lived sessions

Lateral Movement Detection Summary

Technique → Key Event

Pass-the-Hash: Event 4624 Type 3, NTLM auth, unexpected source IP

Pass-the-Ticket: Event 4769 from wrong workstation

Golden Ticket: Event 4672 for domain admin on unexpected system

Kerberoasting: Event 4769 RC4 EncType 0x17 in bulk from one source

DCSync: Event 4662 replication rights exercised by non-DC account

PsExec: Event 7045 PSEXESVC service install on target host

Technique → Key Event (cont.)

Scheduled Task: Event 4698 task created on remote host

WMI execution: wmiprvse.exe spawning cmd/powershell (Event 4688)

RDP: Events 4778/4779 on target system

WinRM: wsmprovhost.exe as parent process (Sysmon Event 1)

SSH tunnel: long-lived SSH sessions with low byte rates

Credential dump: Sysmon Event 10 — lsass.exe accessed by non-system process

Lateral Movement Remediation Steps

Contain: isolate compromised systems from the network — stop active lateral movement

Identify scope: determine all systems the attacker accessed using event log correlation

Reset KRBTGT TWICE: invalidates all forged Golden Tickets (wait >10 hours between resets)

Reset compromised accounts: change passwords for all accounts whose hashes were dumped

Enable Credential Guard: prevents future LSASS credential dumping on Windows 10/11

Deploy LAPS: randomize local admin passwords — prevents PtH reuse across machines

Enable Protected Users security group: prevents NTLM auth and credential caching for admins

Review and remediate SPNs: remove orphaned accounts, enforce AES-only Kerberos

Preventing Lateral Movement — Architecture

Network segmentation: workstations should not communicate directly with other workstations

East-west filtering: ACLs between server VLANs — not just perimeter north-south rules

Privileged Access Workstations (PAWs): dedicated admin systems with no internet access

Tiered admin model: Tier 0 accounts only log into DCs, Tier 1 into servers, Tier 2 workstations

SMB hardening: disable SMBv1 everywhere; require SMB signing (prevents relay attacks)

Disable NTLM where possible: enforce Kerberos-only in domain

Just-in-Time (JIT) access: elevate privileges on demand, expire after task

Zero Trust: verify identity and device posture for every resource access, even inside the network

Conclusion

Lateral movement is the phase that converts a single beachhead into full domain compromise

Pass-the-Hash and Pass-the-Ticket let attackers move without ever knowing a plaintext password

Golden Tickets survive all remediations except double-resetting the KRBTGT account

Kerberoasting is trivially easy — every service account with a weak password is a target

Each remote execution method leaves distinct artifacts — know the evidence trail for each

East-west network segmentation is the single most effective control against lateral movement

Knowledge Check