Bezpieczeństwo infrastruktury krytycznej i systemów sterowania przemysłowego IoT – Laboratorium 1 - Hardening

Forma: praca w samodzielna / małe zespoły (2 - 4 osoby) - opracowane zadania należy przesłać w formie sprawozdania (zadanie z opisem pojawi się na MsTeams)

Wstęp teoretyczny

Windows

Systemy Windows pozostają dominującą platformą dla serwerów i stacji roboczych w obszarze automatyki przemysłowej. Na Windowsie uruchamiane są systemy SCADA i HMI, serwery historyczne, liczne middleware do komunikacji (OPC Classic/UA, Modbus/TCP gateways, oprogramowanie inżynierskie) oraz usługi backendowe, takie jak kontrolery domeny Active Directory, WSUS czy narzędzia dystrybucji oprogramowania. Z punktu widzenia architektury Purdue, elementy te zlokalizowane są zwykle na poziomach 2–3 (sterowanie i operacje), a wybrane funkcje pośredniczące i usługi aktualizacyjne w DMZ 3.5. Odpowiedzialne wdrożenie Windows w OT wymaga pogodzenia wymagań deterministycznej produkcji z cyklem życia bezpieczeństwa systemów IT, dlatego nacisk kładzie się na stabilność, przewidywalność zmian i kontrolę powierzchni ataku, a nie jedynie na maksymalną nowoczesność.

Rola Windows w łańcuchu technologii

Na poziomie sterowania Windows obsługuje interfejs operatora i logikę wizualizacyjną (HMI/SCADA), która komunikuje się z PLC i RTU poprzez protokoły przemysłowe. W warstwie operacyjnej gromadzone są i udostępniane dane historyczne, realizowane są raporty jakości i energii, a na stacjach inżynierskich utrzymywane projekty PLC/HMI. Poza warstwą OT, lecz w jej bezpośrednim sąsiedztwie, działają serwery usługowe, np. repozytoria aktualizacji (WSUS), serwery licencyjne oraz kontrolery domeny dedykowane dla strefy przemysłowej. Ścisła separacja tych ról w topologii stref i kanałów jest kluczowa, gdyż wiele aplikacji przemysłowych wciąż opiera się na starszych komponentach COM/DCOM czy starszych stosach TLS, które źle tolerują agresywne polityki bezpieczeństwa.

System i usługi

Domyślna instalacja Windows zawiera szereg usług, które w OT należy świadomie skonfigurować. Usługa Zdalnego Pulpitu może być używana do nadzoru, ale powinna być osadzona za bramą RDP lub bastionem w DMZ i wymagać NLA, MFA i TLS. Stos SMB powinien mieć wyłączoną wersję v1, a podpisywanie i szyfrowanie SMB należy włączać zgodnie z możliwościami aplikacji. Mechanizmy uwierzytelniania wymagają ograniczenia NTLM i preferowania Kerberosa, w strefach z kontrolerami domeny dla OT wdraża się własną, odseparowaną hierarchię AD, tak aby kompromitacja IT nie implikowała natychmiastowej eskalacji w OT. Lokalna polityka bezpieczeństwa, UAC, zasady haseł oraz blokady kont mają być spójne z zasadą najmniejszych uprawnień, konta serwisowe aplikacji przemysłowych otrzymują precyzyjnie zdefiniowane uprawnienia. Synchronizacja czasu ma znaczenie operacyjne i dochodzeniowe, preferowane jest NTP w całej strefie oraz, tam gdzie wymagane są precyzyjne znaczniki procesu, PTP na poziomie urządzeń czasu rzeczywistego. Usługi logowania są ustawiane tak, by rejestrować zdarzenia bezpieczeństwa (logon/logoff, zmiany uprawnień, uruchomienia procesów, komunikację zdalną). W praktyce rozszerza się możliwości natywnego dziennika poprzez Sysmon i telemetrykę ETW, co pozwala korelować zdarzenia z zakresu ICS w narzędziach SIEM. W wielu zakładach stosuje się kompromis: pełne EDR w DMZ i na serwerach pomocniczych, a na stacjach HMI/SCADA antywirus z precyzyjnymi wyjątkami i monitorowaniem integralności, aby nie kolidować z aplikacją czasu rzeczywistego.

Hardening i kontrola powierzchni ataku

W OT hardening systemu rozpoczyna się od minimalizacji obrazu - usuwa się nieużywane funkcje i aplikacje, blokuje zbędne usługi (w szczególności RemoteRegistry, Fax, niepotrzebne komponenty IIS), ogranicza interfejsy i protokoły do ścisłego minimum. Stosuje się zasady Group Policy, ale w wariancie przewidywalnym dla aplikacji przemysłowych, bazuje się na szablonach Microsoft Security Baselines oraz profilach CIS, weryfikując każdą zmianę w środowisku testowym. Wymogiem staje się whitelisting plików wykonywalnych: WDAC lub w starszych wersjach AppLocker, co pozwala blokować uruchomienia spoza zestawu zatwierdzonych binarek i katalogów. W obszarach szczególnie krytycznych wdraża się tryby kiosku, blokady powłoki i polityki ograniczonego języka PowerShell, a interakcje z systemem ogranicza się do dedykowanych kont operatorskich. Kontrola urządzeń wymiennych obejmuje blokady USB lub dopuszczenie jedynie sprzętu klasy trusted media z podpisami i skanowaniem w strefie kwarantanny.

Aktualizacje, zgodność i zarządzanie zmianą

W przeciwieństwie do IT, w OT priorytetem jest stabilność procesu. Aktualizacje są dystrybuowane w pierścieniach, po pozytywnych testach w bliźniaczym środowisku i o ile wymagają tego dostawcy SCADA/PLC po uzyskaniu akceptacji vendorów. Stosuje się Windows Server Update Services ulokowany w DMZ z repliką pakietów z Internetu poprzez łańcuch pośredników, stacje HMI łączą się tylko do wewnętrznego WSUS. Krytyczne poprawki bezpieczeństwa, które nie mogą być natychmiast zastosowane, wymagają kontroli kompensacyjnych: zawężenia reguł firewall, dodatkowego monitoringu i czasowych ograniczeń dostępu zdalnego. Zmiany konfiguracyjne podlegają procesowi MOC: dokumentuje się powód, zakres, plan odwrotu i wyniki testów, a ich implementację odnotowuje system ticketowy (np. SNOW). Kopie zapasowe zarówno danych procesowych, jak i konfiguracji aplikacji, obrazów systemów oraz projektów inżynierskich – przechowywane są offline i okresowo odtwarzane w trybie ćwiczeń, ponieważ tylko test przywracania weryfikuje realną gotowość.

Tożsamość, domeny i zasada warstwowej administracji

W wielu zakładach wdraża się osobną domenę AD dla OT, z zaufaniem jednokierunkowym do domeny IT lub całkowitym brakiem zaufania i federacją przez bastiony (jumphosty). Model administracji warstwowej ogranicza przywileje: konta Tier-0 zarządzają wyłącznie kontrolerami domeny i systemami tożsamości, a praca administratorów odbywa się z dedykowanych stacji. Wymagane są mechanizmy MFA i uprzywilejowane sesje przez PAM/PSM, tak aby dostęp do serwerów HMI/SCADA odbywał się wyłącznie przez kontrolowane jumpy. Lokalna administracja jest eliminowana, gdy jest niezbędna, wykorzystywany bywa LAPS do rotacji haseł i ścisła ewidencja użycia.

Specyfika aplikacji przemysłowych i kompatybilność

Wiele aplikacji OT korzysta z COM/DCOM i starszych bibliotek. Utwardzanie DCOM wprowadzone globalnie w systemach Windows wymagało aktualizacji komponentów OPC Classic. Z tego powodu każda zmiana w warstwie kryptograficznej (wyłączenie starszych szyfrów, wymuszenie TLS 1.2/1.3, FIPS mode) oraz w politykach DCOM powinna być kwalifikowana w laboratorium z udziałem dostawcy rozwiązania. W komunikacji po SMB stosuje się podpisywanie i izolację udziałów, ale tam, gdzie starszy komponent tego nie wspiera, preferuje się kapsułkowanie ruchu w tunelach przez bastion zamiast otwierania portów w poprzek stref. W przypadku protokołów automatyki działających poprzez Windows (np. bramki Modbus/TCP, IEC-104, DNP3) obowiązuje zasada najmniejszego zaufania: procesy gateway działają na kontach bez uprawnień administracyjnych, a zapora lokalna dopuszcza jedynie precyzyjne adresy i porty.

Monitorowanie i reagowanie

System Windows udostępnia bogaty materiał dowodowy. Kluczowe ślady obejmują zdarzenia logowania (4624, 4625), eskalację uprawnień i modyfikacje grup (4728–4732), uruchamianie procesów (4688 z włączonymi liniami poleceń), zdalne sesje RDP i SMB, a także wpisy Sysmon o operacjach plików i sieci. Zbieranie logów powinno być odseparowane od strefy produkcyjnej, agent kolektor trafia do DMZ, skąd dane są przekazywane do centralnego SIEM. W praktyce łączy się to z pasywną telemetrią NDR na granicach stref, aby wykrywać anomalie, np. nietypowe odpytywanie HMI przez hosty spoza listy białej lub ruch DCOM poza dozwolonymi kanałami. Plan reakcji zakłada izolację logiczną systemu (odłączenie interfejsu uplink, zablokowanie kont w AD OT), utrzymanie procesu w trybie awaryjnym oraz ścieżkę eskalacji do zespołu procesowego i producenta aplikacji.

Windows Security

Tożsamość i uwierzytelnianie w domenie

W domenie AD trzonem jest Kerberos. W konfiguracji produkcyjnej eliminuje się, włącza podpisywanie i kanał zabezpieczony LDAP, wymusza nowoczesne typy szyfrowania biletów (AES-256), wyłącza RC4 i DES oraz aktywuje Kerberos FAST (armoring) i PAC signature. Konta o wysokim ryzyku umieszcza się w „Protected Users" i wiąże z Authentication Policies, aby ograniczyć miejsca i czas logowania. Konta usługowe przenosi się na gMSA, co automatyzuje rotację kluczy i wiąże tożsamość procesu ze SPN bez przechowywania haseł. Zarządzanie lokalnymi administratorami na stacjach i serwerach realizuje się przez Windows LAPS, który wymusza unikatowe, rotowane hasła i bezpieczny escrow w AD. Model nadawania uprawnień opiera się na zasadzie AGDLP oraz podejściach just-enough i just-in-time.

Polityki grupy i hardening konfiguracji

GPO to mechanizm egzekwowania konfiguracji. W praktyce rozdziela się ustawienia komputerów i użytkowników, wiąże je do OU zgodnie z modelem delegacji oraz stosuje „loopback processing" na hostach specjalnego przeznaczenia (np. kiosk/HMI). W obszarze zabezpieczeń systemowych wymusza się: NLA i TLS dla RDP, Remote Credential Guard lub tryb Restricted Admin dla sesji uprzywilejowanych, podpisywanie i szyfrowanie SMB oraz całkowite wyłączenie SMBv1; polityki blokady kont i złożoności haseł, wymaganie kart inteligentnych lub FIDO2 tam, gdzie to uzasadnione. Na kontrolerach domeny i serwerach LDAP egzekwuje się LDAP signing i channel binding. Dodatkowo ogranicza się usługi zbędne, stosuje restrykcje instalacji urządzeń i sterowników, wymusza automatyczne aktualizacje ustawiane w pierścieniach testowych, a konfigurację traktuje jak kod (DSC/Intune/ConfigMgr).

Izolacja poświadczeń i integralność kodu

Obrona przed atakami na LSASS i jądro opiera się na izolacji w oparciu o wirtualizację. Włącza się Credential Guard, LSASS jako proces chroniony (RunAsPPL) oraz HVCI/Kernel-mode Code Integrity wymuszające podpisywanie sterowników. Secure Boot i weryfikacja integralności łańcucha rozruchowego blokują uruchamianie nieautoryzowanych komponentów. Te mechanizmy są filarem oporu przeciwko kradzieży poświadczeń i technikom typu BYOVD.

Kontrola aplikacji i powierzchni ataku

Kontrola wykonywania kodu realizowana jest dwutorowo. AppLocker pozwala dopuszczać binarki według wydawcy, ścieżki i skrótu dla plików EXE, MSI, skryptów i AppX, wdrożenie prowadzi się od trybu audytu do egzekwowania. W środowiskach wymagających silniejszych gwarancji stosuje się Windows Defender Application Control (WDAC), który bazuje na politykach integralności kodu i może być wzmocniony HVCI. Powierzchnię ataku redukuje się regułami ASR (blokada procesów potomnych z Office, wykonywania kodu z pamięci, wstrzygnięć w LSASS, nadużyć WMI), Controlled Folder Access przeciw ransomware oraz AMSI i SmartScreen, które dają wgląd i skanowanie treści skryptowych i pobieranych plików. W pakiecie Office egzekwuje się blokadę makr z internetu (Mark-of-the-Web), dopuszcza jedynie makra podpisane i wyłącza tryby zaufania użytkownika.

Ochrona urządzeń końcowych: AV/EDR, zapora, IPSec

Wbudowany Microsoft Defender Antivirus z włączoną ochroną przed manipulacją stanowi standard. Po podłączeniu do Defender for Endpoint uzyskuje się funkcje EDR: detekcję behawioralną, korelację zdarzeń, izolację hosta i automatyczne działania naprawcze. Zapora hostowa musi pozostać aktywna na wszystkich profilach, a komunikację między segmentami warto wiązać IPSec (izolacja domenowa, IKEv2, certyfikaty z AD CS), co zapewnia wzajemne uwierzytelnienie i prywatność ruchu.

Rejestrowanie i obserwowalność

Zaawansowane zasady audytu włączają szczegółowe kategorie (logowania, użycie uprawnień, dostęp do obiektów, modyfikacje zasad, zdarzenia katalogowe). Zdarzenia są centralizowane przez Windows Event Forwarding do kolektora, a dalej do SIEM. Sysmon uzupełnia dzienniki systemowe o procesy, łącza sieciowe, modyfikacje rejestru i sterowników. PowerShell ma aktywne script block logging i transkrypcje; w środowiskach szczelnych egzekwuje się Constrained Language Mode. Taki łańcuch rejestrowania zapewnia materiał dowodowy i bazę do detekcji anomalii.

Szyfrowanie danych i integralność urządzeń

BitLocker z TPM+PIN szyfruje stacje i serwery, a klucze odzyskiwania są automatycznie deponowane w AD. Polityki obejmują Network Unlock tam, gdzie potrzebny jest bezobsługowy rozruch, oraz wymóg ochrony nośników wymiennych. EFS wykorzystywany jest selektywnie; preferowane jest szyfrowanie całych woluminów i kontrola dostępu do plików przez NTFS/SMB.

Drukowanie, nośniki, urządzenia peryferyjne

Po incydentach typu PrintNightmare twardnieje się Point-and-Print, dopuszcza wyłącznie podpisane sterowniki i ogranicza aktualizacje z nieufnych źródeł. Instalację urządzeń peryferyjnych ogranicza się klasami ID, pozwalając tylko na autoryzowane sprzęty. Dostęp do USB bywa policyjnie wyłączony, a dla wyjątków stosuje się listy dopuszczonych urządzeń i szyfrowanie zawartości.

Zarządzanie poprawkami i konfiguracją

WSUS, Configuration Manager lub Intune realizują plan „pierścieni" aktualizacji z buforem testowym i automatycznym raportowaniem zgodności. Drivery aktualizuje się kontrolowanie. Konfiguracja jest powtarzalna i wersjonowana (GPO jako artefakty, DSC jako deklaracje), a zmiany przechodzą proces akceptacji. Do kontroli zgodności stosuje się baseline'y Microsoft Security Baselines/CIS i regularne skany konfiguracji.

Linux

Linux jest jedną z kluczowych platform w warstwach sterowania i operacyjnej infrastruktury przemysłowej. Jego największe atuty to przewidywalność i możliwość zbudowania minimalnego obrazu ograniczonego do absolutnych potrzeb. W obszarze integracji system oferuje dojrzały stos sieciowy oraz bogaty zestaw usług: NTP/chrony i PTP (ptp4l, phc2sys), brokery MQTT, serwery OPC UA czy lekkie mechanizmy orkiestracji kontenerów dla funkcji brzegowych.

Bezpieczna eksploatacja zaczyna się od minimalnego obrazu. Instalujemy wyłącznie wymagane pakiety, zbędne demony (np. Avahi, CUPS, nieużywane komponenty poczty) pozostają wyłączone. Dostęp sieciowy podlega zaporze nftables lub firewalld, dopuszczone są tylko precyzyjnie określone porty i źródła. Zdalny dostęp zapewnia wyłącznie OpenSSH z wyłączonym logowaniem hasłowym, kluczami publicznymi, regułami Match (adresy, grupy), wymuszonym poleceniem startowym i gdzie to potrzebne uwierzytelnianiem wieloskładnikowym przez PAM. Rola bastionu w DMZ to centralny punkt dostępu administracyjnego; bezpośrednie połączenia z sieci IT do hostów OT są niedozwolone, a tunelowanie i przekierowania portów wyłączone w konfiguracji sshd.

Zarządzanie uprawnieniami opiera się na zasadzie najmniejszego przywileju. Uprawnienia administracyjne nadaje się przez sudo, z dokładnie zdefiniowanymi poleceniami i pełnym logowaniem. SELinux lub AppArmor w trybie enforcing zapewniają separację na poziomie MAC. W środowiskach krytycznych przygotowuje się dedykowane profile dla usług (OPC UA, MQTT, bramki protokołów), aby ograniczyć skutki ewentualnego naruszenia. Integralność i dopuszczanie tylko zaufanego kodu wzmacniają IMA/EVM, dm-verity, OSTree oraz niemutowalne dystrybucje - ograniczają trwałe modyfikacje, co ułatwia audyt.

Aktualizacje wymagają procesu podobnego do tego z Windows, ale z uwzględnieniem specyfiki dystrybucji. Repozytoria są mirrorowane wewnętrznie; zmiany przechodzą przez środowisko testowe i są wdrażane w pierścieniach. Gdy wymagana jest zgodność ze starszymi wersjami oprogramowania SCADA, stosuje się wersjonowanie i pinning pakietów. W węzłach bezdyskowych lub z ograniczonym dostępem sieciowym używa się repozytoriów offline i podpisanych pakietów; klucze GPG dla repozytoriów OT przechowuje się w bezpiecznym module, a publikację poprzedza weryfikacja sum kontrolnych i próbna instalacja. Mechanizmy live patchingu jądra (kpatch/ksplice) stosuje się tylko tam, gdzie dostawca aplikacji to dopuszcza; w pozostałych przypadkach priorytetem jest przewidywalny restart w zaplanowanym oknie serwisowym.

Rejestrowanie i obserwowalność powinny być spójne w skali strefy. journald stanowi podstawowe źródło logów; rsyslog lub syslog-ng przesyłają je do kolektorów w DMZ, skąd trafiają do SIEM. auditd pozwala śledzić zdarzenia bezpieczeństwa (wywołania, modyfikacje konfiguracji, zmiany uprawnień). W środowiskach o podwyższonym rygorze stosuje się reguły eBPF i narzędzia takie jak Falco do wykrywania anomalii na poziomie jądra i kontenerów. Dobrą praktyką jest rotacja logów, przechowywanie kopii na nośnikach WORM oraz synchronizacja czasu (chrony/PTP), aby zachować wartość dowodową zapisów.

Konteneryzacja naturalnie wpisuje się w Linuksa: Podman lub Docker, separacja przez przestrzenie nazw, cgroups i seccomp, a także etykiety SELinux dla izolacji wolumenów. W OT stosuje się konserwatywne podejście: minimalne, podpisane obrazy skanowane pod kątem CVE; kontenery uruchamiane w trybie rootless, z ograniczonym zestawem capabilities i bez dostępu do gniazda demonów kontenerowych. Ten model ułatwia dystrybucję bramek protokołów i komponentów IIoT, ale wymaga tej samej dyscypliny wersjonowania i testów co klasyczne pakiety.

Linux Security

Bezpieczeństwo w Linuksie to efekt współdziałania wielu warstw: tożsamości i uwierzytelniania, twardnienia hosta, kontroli wykonywania kodu, dyscypliny sieciowej, rzetelnego logowania oraz przewidywalnych aktualizacji. W ICS/OT te warstwy muszą dodatkowo respektować segmentację według modelu Purdue.

Tożsamość. W OT Linuksa zwykle dołącza się do odseparowanej domeny tożsamości (SSSD/realmd z Kerberosem), co pozwala na imienne konta i role. PAM egzekwuje jakość haseł, blokady po nieudanych próbach, polityki sesji i limity zasobów; administracja korzysta z logowania wieloskładnikowego (FIDO2, karty, TOTP). Konto root nie służy do pracy interaktywnej, polecenia uprzywilejowane wykonuje się przez sudo z precyzyjnymi regułami i śladem audytowym. Zdalna administracja odbywa się wyłącznie przez SSH, bez haseł, z kluczami i listą dozwolonych źródeł; sesje inicjuje się z bastionu w DMZ, bez tunelowania i X-forwardingu.

System operacyjny. Obraz systemu powinien być minimalny i przewidywalny. Nad klasycznym modelem uprawnień (DAC) nakłada się kontrolę dostępu obowiązkowego: AppArmor lub SELinux w trybie egzekwowania. Usługi uruchamia się w piaskownicach systemd: pod nieuprzywilejowanym użytkownikiem, z prywatnym katalogiem tymczasowym, systemem plików w trybie tylko do odczytu z wąskimi wyjątkami, z wyłączeniem możliwości nadawania nowych przywilejów, z ograniczonym zestawem capabilities i filtrem syscalls (seccomp). Dzięki temu kompromitacja jednego procesu nie daje kontroli nad całym hostem.

Jądro i sieć. Parametry sysctl (ASLR, ochrona wskaźników, restrykcje ptrace, wyłączenie przekierowań i akceptowania redirectów) ograniczają skuteczność technik post-exploitation. Zapora hostowa na nftables działa w trybie domyślnego odrzucania i dopuszcza tylko ściśle określony ruch, najlepiej skojarzony z tożsamością kanału administracyjnego (np. IPSec). W sieci reguły są jasno pozytywne: brak zbędnej komunikacji, brak mostów między strefami, brak hostów z dwoma interfejsami w różnych domenach. Spójny czas zapewnia chrony, a tam, gdzie to potrzebne, PTP - to warunek wiarygodnych logów i analiz powłamaniowych.

Wykonywanie kodu i integralność. Poza piaskownicami warto ograniczyć, co w ogóle może się uruchamiać. IMA/EVM wiąże pliki wykonywalne i konfiguracje z podpisami lub skrótami weryfikowanymi przez kernel. AIDE jako system kontroli integralności okresowo porównuje stan hosta z zaufaną bazą. W bramkach i węzłach brzegowych rośnie popularność dystrybucji niemutowalnych z aktualizacjami atomowymi i łatwym rollbackiem. Repozytoria są mirrorowane wewnętrznie i podpisane kluczami GPG; wersje komponentów są przypinane po testach, aby unikać niekontrolowanych aktualizacji zależności.

Obserwowalność. journald i rsyslog przekazują logi do kolektora w DMZ, dalej do SIEM. auditd rejestruje tworzenie procesów, zmiany uprawnień i modyfikacje konfiguracji; narzędzia oparte na eBPF (Falco) czy Sysmon for Linux dodają detekcję zachowań na poziomie wywołań systemowych. Logi mają wartość tylko wtedy, gdy czas jest spójny, a łańcuch przesyłu stabilny i odporny na manipulacje - dlatego przepływ zawsze przebiega przez DMZ.

Aktualizacje. W OT to proces, nie jednorazowa akcja. Pakiety pobiera się z wewnętrznych mirrorów, zmiany przechodzą przez pierścienie (laboratorium → pre-prod → produkcja) i okna serwisowe. W razie krytycznej luki stosuje się zabezpieczenia kompensacyjne do czasu zakończenia testów: ciaśniejsze reguły zapory, podniesienie czułości detekcji, wyłączenie nieużywanych interfejsów. Konfigurację traktuje się jak kod (Ansible, Salt), dzięki czemu da się odtworzyć host deterministycznie, a różnice między stanem deklarowanym i rzeczywistym są mierzalne. Węzły offline korzystają z podpisanych repozytoriów i kontrolowanych procedur wnoszenia paczek.

Szyfrowanie i kopie zapasowe. LUKS2 zabezpiecza dane w spoczynku. W serwerowniach można rozważyć TPM lub zdalne odblokowanie powiązane ze strefą OT; w terenie lepsze są hasła i procedury z kontrolą czterech oczu. Kopie konfiguracji, danych i obrazów wykonuje się z szyfrowaniem i weryfikacją integralności, a przywracanie testuje regularnie - tylko udany test potwierdza realne RTO/RPO. W repozytoriach konfiguracji przechowuje się także polityki SELinux/AppArmor, pliki unit systemd, klucze repozytoriów i skrypty wdrożeniowe.

Kontenery i edge. Preferowany jest Podman w trybie rootless lub Docker z mapowaniem przestrzeni użytkownika. Kontenery działają na systemach plików tylko do odczytu, z ograniczonymi przywilejami, profilami seccomp i etykietami SELinux/AppArmor. Montowanie gniazd demonów i tryb uprzywilejowany są wykluczone. Obrazy muszą być podpisane i skanowane pod kątem CVE, a ich SBOM archiwizowany - to warunek szybkiej reakcji na luki w łańcuchu dostaw.

Materiały dodatkowe:

- https://www.cisecurity.org/cis-benchmarks
- https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/windows-security-configuration-framework/windows-security-baselines
- https://adsecurity.org/
- https://support.microsoft.com/en-us/windows/firewall-and-network-protection-in-the-windows-security-app-ec0844f7-aebd-0583-67fe-601ecf5d774f
- https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-overview
- https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/applocker/applocker-overview
- https://www.jakubkulikowski.pl/2020/08/27/kontrola-oprogramowania-applocker/
- https://helion.pl/ksiazki/mastering-defensive-security-effective-techniques-to-secure-your-windows-linux-iot-and-cloud-inf-cesar-bravo-darren-kitchen,e_2t2t.htm?srsltid=AfmBOor89hfUM54LbBzSg21jdMGcGQ3_gjZsGbA6iC45SvLhd1hItlWX#format/e
- https://dev.to/odoth4kz/hardening-the-home-lab-5fm1

Przykłady

Windows

W0. Sprawdzanie uprawnień

1. Otwórz PowerShell jako **zwykły** użytkownik.
2. Uruchom:

```
whoami
whoami /groups
whoami /priv
net user %USERNAME%
Get-LocalGroupMember Administrators
Get-LocalGroup | ForEach-Object { "$($_.Name):"; (Get-LocalGroupMember $_.Name).Name }
```

Co w ten sposób zobaczymy?

nazwę konta, listę grup (np. *Users*, *Administrators*, *Remote Desktop Users*), listę przywilejów (Se*), właściwości konta.

W1. Uprawnienia systemowe (User Rights Assignment)

1. Otwórz **secpol.msc** → *Local Policies* → *User Rights Assignment*.
2. Sprawdź szczególnie:
    - **Deny log on locally / via RDP**
    - **Log on as a service / batch job**
    - **Back up files and directories**
    - **Debug programs**
    - **Load and unload device drivers**
3. Poszukaj innych ciekawych uprawnień

Jak ograniczyć:

- Usuń zbędne grupy/konta z tych praw
- Dodaj „Deny” dla kont, które nie powinny się logować lokalnie/RDP

W2. Członkostwo w grupach i RDP

1. Sprawdź członkostwo:

```
Get-LocalGroupMember Administrators
Get-LocalGroupMember "Remote Desktop Users"
```

2. Jeśli nie musisz być administratorem, zdejmij siebie z grupy:

```
Remove-LocalGroupMember -Group Administrators -Member $env:USERNAME
```

3. RDP tylko dla wybranych:

```
net localgroup "Remote Desktop Users" /delete %USERNAME%  # jeśli nie potrzebujesz
```

W3. Uprawnienia do plików/katalogów (NTFS/ACL)

1. Sprawdź efektywne uprawnienia do np. C:\Tools:

```
icacls C:\Tools
```

2. Usuń dziedziczenie i zawęź dostęp:

```
icacls C:\Tools /inheritance:d
icacls C:\Tools /remove:g "Users"
icacls C:\Tools /grant:r "$env:USERNAME:(OI)(CI)M"
```

3. Jeśli tworzysz konto-serwis, daj mu **tylko** *RX*:

```
icacls C:\Tools\MySvc /grant "SVC_MyApp:(OI)(CI)RX"
```

W4. Autostart i usługi z nadmiernymi prawami

1. Sprawdź programy startowe:

```
Get-CimInstance Win32_StartupCommand | Select Name, Command, Location
```

2. Sprawdź usługi działające jako **LocalSystem**:

```
Get-WmiObject Win32_Service | where {$_.StartName -eq "LocalSystem"} | Select Name, State, StartMode
```

3. Zmień konto usługi na dedykowane o minimalnych prawach (GUI: *services.msc* → *Log On*).
   Alternatywnie PowerShell:

```
$svc = Get-WmiObject Win32_Service -Filter "Name='MyService'"
$svc.Change($null,$null,$null,$null,$null,$null,".\SVC_MyApp","Haslo#Silne123")
```

W5. Rollback

- Zanim ograniczysz prawa, utwórz punkt przywracania: SystemPropertiesProtection.exe.
- Eksport polityk lokalnych: secedit /export /cfg C:\hardening\secpol.inf.

W6. Badanie miskonfiguracji uprawnień

Pobierz i uruchom winpeas.bat / winpeas.exe: https://github.com/peass-ng/PEASS-ng/releases/tag/20251004-ba856a2a

Zaobserwuj działanie programu 😊

Linux

L0. Sprawdzanie uprawnień

1. W terminalu:

```
id
whoami
groups
sudo -l
getent passwd $USER
```

2. Procesy działające jako root:

```
ps -U root -u root u | head -n 20
```

L1. Członkostwo i sudo

1. Sprawdź, czy jesteś w sudo:

```
getent group sudo
```

2. Jeżeli pełne sudo nie jest wymagane, **usuń** siebie z grupy:

```
sudo gpasswd -d $USER sudo   # Debian/Ubuntu
```

3. Utwórz **ograniczone** reguły sudo tylko do wybranych poleceń:

```
sudo visudo -f /etc/sudoers.d/10-minimal
# w pliku dodaj (przykład):
# Cmnd_Alias NETTOOLS = /usr/sbin/ufw status, /usr/sbin/ufw allow *, /usr/sbin/ufw deny *
# %netops ALL=(root) NOPASSWD: NETTOOLS
```

4. Dodaj siebie do grupy netops **zamiast** do sudo:

```
sudo groupadd -f netops
sudo usermod -aG netops $USER
```

L2. SSH i logowanie

1. Sprawdź konfigurację:

```
sshd -T | egrep 'permitrootlogin|passwordauthentication|maxauthtries|allowusers|port'
```

2. Ogranicz logowanie:

    - Otwórz /etc/ssh/sshd_config:

```
PermitRootLogin no
PasswordAuthentication no
AllowUsers <twoj_user>
MaxAuthTries 4
```

- Wygeneruj klucze: ssh-keygen -t ed25519.
- Restart: sudo systemctl restart ssh.

L3. Uprawnienia do plików/directoriów

1. Sprawdź umask:

```
umask
```

2. Ustaw szczelniejsze domyślne prawa (np. 077 dla kont serwisowych):

- Dla sesji: umask 077
- Systemowo (Debian/Ubuntu): edytuj /etc/login.defs (UMASK 077) lub /etc/pam.d/common-session* z umask.

3. Zawężanie w istniejących katalogach:

```
chmod 750 /opt/myapp
chown -R svc_myapp:svc_myapp /opt/myapp
setfacl -m u:$USER:rx /opt/myapp/logs
getfacl /opt/myapp > ~/hardening/facl.txt
```

L4. SUID/SGID i capability

1. Znajdź pliki z **SUID/SGID**:

```
sudo find / -xdev \( -perm -4000 -o -perm -2000 \) -type f -print 2>/dev/null | tee ~/hardening/suid_sgid.txt
```

2. Oceń, czy są potrzebne. Zdejmij bit SUID tam, gdzie nie jest konieczny:

```
sudo chmod u-s /usr/local/bin/stary_program
```

3. Sprawdź **Linux capabilities**:

```
sudo getcap -r / 2>/dev/null | tee ~/hardening/caps.txt
```

4. Usuń zbędne capability:

```
sudo setcap -r /usr/local/bin/stary_progra
```

L5. Uprawnienia usług (systemd)

1. Zidentyfikuj usługi działające jako root:

```
systemctl --type=service --state=running | awk '{print $1}' | xargs -I{} sh -c 'systemctl show {} -p User -p Group -p FragmentPath'
```

2. Nadawaj minimalne uprawnienia w pliku *.service* (override):

```
sudo systemctl edit myapp.service
# dodaj:
[Service]
User=svc_myapp
Group=svc_myapp
NoNewPrivileges=true
CapabilityBoundingSet=
PrivateTmp=true
ProtectSystem=strict
ProtectHome=true
RestrictAddressFamilies=AF_INET AF_UNIX
```

3. sudo systemctl daemon-reload && sudo systemctl restart myapp.service

L6. Badanie miskonfiguracji uprawnień

Pobierz i uruchom linpeas.sh: https://github.com/peass-ng/PEASS-ng/releases/tag/20251004-ba856a2a

Zaobserwuj działanie programu 😊

Zadania praktyczne

Do wykonania zadań można utworzyć maszynę wirtualną z Windows oraz dowolną dystrubucją linux (zalecany ubuntu/debian).

Zadanie 1 - Hardening Windows

Cel: Zmniejszenie powierzchni ataku Windows Server/różne dystrybuuje Windows poprzez odpowiedni hardening - bezpieczna konfiguracja kont, usług, zapory, logowania, ASR i szyfrowania, polityk grupowych, AD, etc.

https://www.cisecurity.org/cis-benchmarks

https://happycamper84.medium.com/windows-hardening-tryhackme-walkthrough-5fb3ff7aa5c6

Opracuj 5 metod hardeningu windows (można więcej 😊 i można wymyślić swoje spoza listy) znajdź a następnie opisz/wykonaj (w zależności od możliwości) jak można zaimplementować te zabezpieczenia.

Przykładowa lista:

1. Skonfiguruj polityki haseł i blokady konta zgodnie z dobrymi praktykami (min. długość, złożoność, historia, lockout).
2. Wyłącz/ogranicz konto lokalnego administratora: zmień nazwę, ustaw losowe hasło, wdróż LAPS/LAPS-C.
3. Włącz UAC na poziomie „Always notify" i uzasadnij wpływ na bezpieczeństwo i użyteczność.
4. Skonfiguruj BitLocker z TPM i PIN (OS, Fixed, Removable) + odzyskiwanie kluczy w bezpiecznym repozytorium.
5. Włącz Credential Guard oraz skonfiguruj LSASS jako proces chroniony (RunAsPPL).
6. Wymuś SMB podpisywanie i szyfrowanie; wyłącz SMBv1; przeanalizuj wpływ na kompatybilność.
7. Ogranicz NTLM (Audit → Deny) i wymuś Kerberos; zweryfikuj w logach, które aplikacje wciąż używają NTLM.
8. Skonfiguruj Windows Firewall z regułami „deny-by-default" dla przychodzących; dodaj tylko niezbędne wyjątki.
9. Wdróż reguły Microsoft Defender Attack Surface Reduction (ASR) i zademonstruj zablokowanie ryzykownego zachowania (np. child-process z Office).
10. Skonfiguruj Defender AV/EDR: tryb real-time, cloud-delivered, PUA, heurystyka, Controlled Folder Access.
11. Zastopuj i ustaw „Disabled" dla zbędnych usług (np. Fax, XPS) zgodnie z zasadą najmniejszych uprawnień.
12. Włącz Network Level Authentication i ogranicz RDP: niestandardowy port, limit prób, ban po nieudanych logowaniach, wymuś MFA.
13. Skonfiguruj AppLocker lub Windows Defender Application Control (WDAC) w trybie „Allow-list"; zezwól tylko podpisane i firmowe aplikacje.
14. Skonfiguruj Exploit Protection (per-app) na bazie profilu: DEP, ASLR, CFG; porównaj działanie przed/po.
15. Włącz i ustandaryzuj rejestrowanie: Advanced Audit Policy, PowerShell (Module, ScriptBlock, Transcription), Object Access.
16. Zainstaluj i skonfiguruj Sysmon z własnym zestawem reguł; zademonstruj wykrycie zdarzenia (np. nietypowe połączenie z cmd.exe).
17. Wdróż polityki makr Office: blokada makr z internetu, podpisy cyfrowe; sprawdź scenariusz ataku z dokumentem .docm.
18. Włącz Secure Boot i Device Guard; potwierdź integralność łańcucha rozruchu.
19. Skonfiguruj polityki dla nośników wymiennych: blokada auto-run, ograniczenia zapisu/odczytu, szyfrowanie.

20. Utwórz listy kontroli dostępu (NTFS ACL) dla współdzielonego katalogu: rozdziel READ/WRITE, usuń „Everyone".
21. Zbuduj GPO „Baseline" dla stacji roboczych zgodną z Microsoft Security Baselines/CIS i zastosuj ją do OU „Lab-Workstations".
22. Wdróż harmonogram aktualizacji (Windows Update for Business/WSUS), zdefiniuj pierścienie, deadline i restarty.
23. Ogranicz prawa lokalne: usuń użytkowników z „Administrators/Power Users"; zastosuj „Deny log on locally/through RDP" tam, gdzie potrzebne.
24. Włącz i skonfiguruj Windows Defender Credential Guard i Remote Credential Guard dla sesji zdalnych.
25. Zmień domyślne kanały logów na „retention/do not overwrite" i zwiększ ich rozmiar; uzasadnij wartości.
26. Skonfiguruj polityki blokujące „living-off-the-land": ogranicz w GPO dostęp/uruchamianie narzędzi (powershell.exe, wmic, certutil) dla użytkowników nietechnicznych.
27. Włącz i wymuś blokadę ekranu po X minutach, wymaganie hasła po wznowieniu; ustaw politykę wygaszacza.
28. Skonfiguruj polityki drukarek tak, by eliminować vektory „PrintNightmare": restrykcje instalacji sterowników, „Point and Print".
29. Utwórz plan kopii zapasowych (Volume Shadow Copy/robocopy + szyfrowanie) i test odtwarzania pliku po symulacji ataku ransomware.
30. Opracuj skrypt hardeningu (PowerShell DSC/Intune) automatyzujący najważniejsze ustawienia z tej listy; uruchom go i pokaż raport zgodności.
31. Zaimplementuj „Just-Enough Administration" (JEA) dla zadań helpdesk: utwórz sesję PowerShell z ograniczonymi cmdletami.
32. Wdróż politykę „Software Restriction Policies" jako alternatywę dla AppLocker; porównaj efekty i ograniczenia.
33. Zidentyfikuj i usuń „persistence mechanisms" atakującego (Run keys, Scheduled Tasks, WMI, Services); udokumentuj proces.
34. Przeprowadź mini-red-team check: uruchom nieszkodliwy test (np. symulacja dump LSASS) i udowodnij, że wdrożone zabezpieczenia to blokują/logują.
35. Przygotuj raport zgodności z baseline (checklist + wynik skryptu); wskaż odchylenia, ryzyko i plan remediacji.
36. Zbuduj separację ról przez lokalne zasady i GPO (User Rights Assignment): kto może „Debug programs", „Load driver", „Back up files".
37. Skonfiguruj Windows Event Forwarding (WEF) z hostów do kolektora; zaprezentuj centralny przegląd krytycznych zdarzeń bezpieczeństwa.
38. Zabezpiecz PowerShell: egzekwuj ExecutionPolicy, Constrained Language Mode dla kont nietechnicznych; zweryfikuj działanie.

Zadanie 2 - Hardening Linux

Cel: Utwardzenie serwera linuksowego: aktualizacje, konta, SSH, firewall, logowanie, MAC, pliki i jądro. Etc.

https://www.cisecurity.org/cis-benchmarks

https://daniel-schwarzentraub.medium.com/tryhackme-linux-system-hardening-f1b110634dc0

Opracuj 5 metod hardeningu linux (można więcej 😊 i można wymyślić swoje spoza listy) znajdź a następnie opisz/wykonaj (w zależności od możliwości) jak można zaimplementować te zabezpieczenia.

Przykładowa lista

1. Skonfiguruj politykę haseł i blokady konta w PAM (min. długość, złożoność, historia, blokada po X błędach).
2. Wymuś użycie sudo (zamiast su), ogranicz do grupy sudo/wheel, włącz logowanie komend i wymóg TTY.
3. Włącz 2FA dla SSH z PAM (TOTP/Yubikey) dla wybranych użytkowników.
4. Utwardź SSH: zakaz PermitRootLogin, PasswordAuthentication no (klucze), Kex/HostKeyAlgorithms na silne, AllowUsers/AllowGroups.
5. Skonfiguruj zaporę (nftables/ufw/firewalld) w modelu „deny-by-default" z tylko niezbędnymi wyjątkami.
6. Wyłącz i zamaskuj zbędne usługi systemd (drukowanie, avahi, rpcbind, telnetd, itd.).
7. Włącz i skonfiguruj SELinux lub AppArmor: profil „enforcing" dla serwera www i sshd.
8. Włącz auditd z regułami: dostęp do /etc/shadow, modyfikacje sudoers, zmiany uprawnień, ładowanie modułów kernela.
9. Utwardź sysctl: zakaz source routing, włącz rp_filter, syn_cookies, ogranicz ICMP, wyłącz ip_forward (o ile nie router).
10. Zablokuj nieużywane protokoły i moduły kernela (dccp, sctp, rds, tipc) oraz USB storage (jeśli niepotrzebne).
11. Skonfiguruj mount options: /tmp, /var/tmp, /home, /var/log z nodev,nosuid,noexec (gdzie możliwe).
12. Włącz pełne szyfrowanie dysku LUKS dla partycji danych; zasady rotacji i escrow kluczy.
13. Utwórz i wymuś benchmarkową listę pakietów (allow-list/deny-list), usuń kompilatory i narzędzia exploitacyjne z hostów produkcyjnych.
14. Włącz automatyczne aktualizacje bezpieczeństwa (unattended-upgrades/dnf-automatic) i powiadomienia.
15. Skonfiguruj logowanie (rsyslog/journald): persistent storage, limity rozmiaru, retencja, forward do zewnętrznego log-serwera (TLS).
16. Ogranicz uprawnienia wrażliwych plików: /etc/passwd, /etc/shadow, /etc/sudoers, klucze SSH, pliki systemd unit override.
17. Oczyść PATH, wyłącz . w PATH, ustaw bezpieczne umaski (UMASK 027) globalnie.
18. Zablokuj core dumps (limits, fs.suid_dumpable=0) i włącz pełne ASLR (kernel.randomize_va_space=2).
19. Zredukuj capabilities binarek (setcap), usuń zbędne SUID/SGID, zraportuj wszystkie SUID i uzasadnij wyjątki.
20. Odseparuj usługi w systemd (ProtectSystem, ProtectHome, PrivateTmp, NoNewPrivileges, RestrictAddressFamilies, RestrictNamespaces, CapabilityBoundingSet).
21. Utwardź serwer WWW: nagłówki bezpieczeństwa, minimalne ciphers/TLS1.2+, HSTS, izolacja vhostów, mod_security / nginx njs polityki.
22. Włącz Fail2ban/sshd jail z sensownymi progami i listą dozwolonych sieci.
23. Skonfiguruj NTP (chrony/ntpd) z autentykacją i serwerami referencyjnymi; wymuś UTC dla logów.

24. Wprowadź politykę czystych crontabów/anacron: audyt zadań, maile z wynikami, PATH jawny, umask, blokady cron.deny/allow.
25. Wdróż politykę bezpiecznych logowań konsolowych: wyłącz TTY nieużywane, ban root na TTY, ban hasła w GRUB (chroniony hasłem).
26. Utwardź bootloader (GRUB2): hasło na edycję wpisów, parametry jądra minimalizujące powierzchnię ataku.
27. Włącz sudo z powiadomieniami do SIEM: tagowanie poleceń i korelacja z TTY/hostem.
28. Wymuś SSH CA/TrustedUserCAKeys dla zarządzania kluczami, rotacja i odwoływanie kluczy.
29. Utwardź kontenery (Docker/Podman): rootless, no-new-privileges, ograniczenia cgroups, seccomp profil, AppArmor/SELinux enforcing.
30. Zabezpiecz artefakty CI/CD na hoście: katalogi tylko-do-odczytu, GPG verify, podpisy pakietów, minimalne tokeny.
31. Wykryj i usuń mechanizmy persystencji atakującego: ~/.config/systemd/user, crontab, rc.local, skrypty w /etc/profile.d, udev rules, systemd timers.
32. Przygotuj playbook Ansible (albo skrypt bash) automatyzujący najważniejsze ustawienia z tej listy + raport zgodności.
33. Wdróż bezpieczne polityki dla użytkowników: separacja ról, minimalne uprawnienia do plików i gniazd, dedykowane grupy per usługa.
34. Przeprowadź test: symuluj próbę eskalacji (np. ładowanie modułu, SUID abuse, odczyt /etc/shadow, reverse shell) i wykaż, że polityki to blokują i logują.
35. Skonfiguruj alerting (e-mail/webhook) na krytyczne zdarzenia z auditd, sudo, sshd, AIDE - pokaż przykładowy alert i jego triage.
36. Zaimplementuj politykę bezpiecznego korzystania z nośników wymiennych (udev/usbguard): blokada, wyjątki per VID/PID, logowanie.
37. Utwardź serwer baz danych (PostgreSQL/MySQL) na tym hoście: TLS, pg_hba.conf/mysqld bind, role least-privilege, audit log.
38. Utwardź narzędzia programistyczne (Python/Node/Java): blokada pip install --user globalnie, weryfikacja podpisów, mirror repo z allow-listą.
39. Zaprojektuj i zaimplementuj rotację logów (logrotate/journald) z kompresją, podpisem i retencją zgodną z polityką firmy (dowolna).
40. Utwardź środowisko użytkownika: domyślne aliasy bezpieczeństwa (np. alias rm='rm -i'), bezwzględne ścieżki, czyszczenie umask/IFS.
41. Zaimplementuj skan podatności i zgodności (OpenSCAP/lynis) oraz porównaj wynik przed/po hardeningu.
42. Przygotuj plan kopii zapasowych - szyfrowanych, podpisanych, wykonaj próbę odtworzenia i porównaj checksumy.
43. Ogranicz sudoers przez Cmnd_Alias i Runas_Alias – tylko konkretne binarki z pełnymi ścieżkami, zabroń powłok interaktywnych.

Zadanie 3 - Odpowiedz na pytania

1. Dlaczego wyłączenie SMBv1 i RDP dla wszystkich jest sensowne w środowisku domyślnym? Kiedy można zrobić wyjątek i jak go bezpiecznie wdrożyć?
2. ASR w Defenderze: które 2–3 reguły są najbardziej uciążliwe dla użytkowników i jak minimalizować fałszywe alarmy?
3. SSH: dlaczego „PasswordAuthentication no" oraz klucze Ed25519 to lepsza praktyka niż długie hasła?
4. Czym różni się UFW od fail2ban - które ryzyko redukuje każdy z nich?
5. Po co AIDE, skoro mamy backupy? Jakie scenariusze wykryje AIDE, a backup nie?
6. AppArmor/SELinux w trybie enforcing - podaj przykład, jak polityka MAC mogła zablokować eskalację uprawnień.
7. Jak skonfigurować centralizację logów (Windows + Linux) w małej organizacji (kilka zdań, narzędzia/open-source mile widziane)?
8. Jakie trzy wskaźniki/metryki w logach/audytach monitorować stale, aby szybko wykryć atak?