Problemy

Inwentarz, monitoring w Zabbixie i diagnoza odrzuconego certyfikatu.

- Inwentarz certyfikatów i częste problemy: wygasła CRL, łańcuch, SAN, alarm w Zabbixie

Inwentarz certyfikatów i częste problemy: wygasła CRL, łańcuch, SAN, alarm w Zabbixie

Poziom: średnio

Dotyczy: Windows Server i Windows 11 (magazyny certyfikatów, CAPI2), Rocky Linux 9 (OpenSSL 3), Zabbix 7.0 na mon01 z agentem 2, szablon „Website certificate by Zabbix agent 2”, GLPI, ca01, CA-Zielona, NPS, fw01

Czas: ~45 min

Certyfikat, o którym nikt nie wie, wygasa zawsze w najgorszym momencie, a lista unieważnień, której nikt nie pilnuje, potrafi wyłączyć Wi-Fi w całym biurze. Ta strona pokazuje, jak zebrać wszystkie certyfikaty Zielonej w jednym spisie, jak pilnować ich i list unieważnień w Zabbixie i jak w kilka minut ustalić, dlaczego klient odrzuca certyfikat. Wracamy tu przy każdym alarmie o certyfikacie i raz w miesiącu przy przeglądzie spisu.

Skróty użyte na tej stronie (15)

Skrót	Co znaczy
ACME	Automatic Certificate Management Environment: protokół, którym Traefik albo certbot automatycznie zamawiają i odnawiają certyfikat w Let's Encrypt, bez ręcznego wniosku
AIA	Authority Information Access: miejsce w certyfikacie, które mówi, skąd pobrać certyfikat urzędu, który go podpisał
CAPI2	CryptoAPI 2: warstwa kryptograficzna Windows, która buduje łańcuchy certyfikatów i sprawdza unieważnienia; jej dziennik pokazuje dokładny powód odrzucenia certyfikatu
CD	Continuous Delivery/Deployment: automatyczne wdrażanie zbudowanego kodu
CDP	CRL Distribution Point: adres w certyfikacie, pod którym urząd publikuje listę unieważnionych certyfikatów
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

Skrót	Co znaczy
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami
JSONPath	Zapis ścieżki, która wskazuje jedną wartość w dokumencie JSON, np. $.value to pole value na samym wierzchu
NPS	Network Policy Server: rola Windows Server obsługująca RADIUS, pośredniczy w sprawdzaniu hasła domenowego przy logowaniu do VPN albo Wi-Fi
PKI	Public Key Infrastructure: infrastruktura klucza publicznego, czyli urząd certyfikacji razem z całym mechanizmem wydawania i odwoływania certyfikatów
SAN	w certyfikatach: Subject Alternative Name, dodatkowe nazwy hosta w certyfikacie; w pamięciach: Storage Area Network, sieć dyskowa
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
URL	Uniform Resource Locator: adres strony lub zasobu
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
Wi-Fi	Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii

Fiszka

Polecenia PowerShella wykonujemy jako administrator na serwerze albo zdalnie przez `Invoke-Command`, polecenia bash na serwerze linuksowym albo na `mon01`. Odczyty są bezpieczne, zmieniają tylko fiszki z Zabbixem i dziennikiem CAPI2.

□□ TYLKO ODCZYT

Certyfikaty komputera wygasające w ciągu 60 dni: Get-ChildItem -ExpiringInDays

```
Get-ChildItem Cert:\LocalMachine\My -ExpiringInDays 60 | Select-Object Subject, Issuer, NotAfter, Thumbprint
```

Parametr `-ExpiringInDays` dostawcy `Cert:` wybiera certyfikaty, którym zostało mniej dni niż podana liczba. Pusty wynik to dobra wiadomość.

□□ TYLKO ODCZYT

Zbiór certyfikatów z wielu serwerów Windows: Invoke-Command

```
$serwery = "NazwaSerwera1","NazwaSerwera2"
Invoke-Command -ComputerName $serwery -ScriptBlock { Get-ChildItem Cert:\LocalMachine\My
| Select-Object Subject, Issuer, NotAfter, Thumbprint, @{n='SAN';e={($_.Extensions |
Where-Object { $_.Oid.Value -eq '2.5.29.17' }).Format(0)}} } | Sort-Object NotAfter |
Export-Csv C:\Temp\certyfikaty-windows.csv -NoTypeInformation -Encoding UTF8
```

Komendy można wykonać jako blok. Zbiera z każdego serwera certyfikaty z magazynu komputera razem z nazwami z SAN i zapisuje jeden plik posortowany po dacie wygaśnięcia. Kolumna `PSComputerName` mówi, z którego serwera pochodzi wiersz.

Przykład (skrót):

```
"Subject","Issuer","NotAfter","SAN","PSComputerName"
"CN=sql01.ad.zielona.example","CN=CA-Zielona, DC=ad, DC=zielona, DC=example","02.12.2026
09:14:02","DNS Name=sql01.ad.zielona.example","sql01"
"CN=vpn.zielona.example, OU=IT, ...","CN=CA-Zielona, DC=ad, DC=zielona,
DC=example","03.10.2027 08:22:00","DNS Name=vpn.zielona.example","rds01"
```

□□ **SAN** = w certyfikatach: Subject Alternative Name, dodatkowe nazwy hosta w certyfikacie; w pamięciach: Storage Area Network, sieć dyskowa

□□ **ZMIENIA SYSTEM**

Certyfikaty w plikach na serwerze linuksowym: openssl x509 -enddate

```
sudo find /etc/pki /etc/ssl /etc/letsencrypt/live /docker/apps -type f \( -name '*.crt'
-o -name '*.pem' -o -name '*.cer' \) 2>/dev/null | while read -r plik; do
  koniec=$(openssl x509 -in "$plik" -noout -enddate 2>/dev/null | cut -d= -f2)
  [ -n "$koniec" ] && printf '%s\t%s\t%s\n' "$(date -d "$koniec" +%F)" "$plik"
"$(openssl x509 -in "$plik" -noout -subject 2>/dev/null)"
done | sort
```

Komendy można wykonać jako blok. Przegląda typowe katalogi z certyfikatami, w tym dane aplikacji w `/docker/apps`, i wypisuje datę wygaśnięcia, ścieżkę i podmiot. Pliki z kluczami prywatnymi albo z kilkoma certyfikatami w jednym pliku pokazują tylko pierwszy certyfikat albo nic; to spis do przeglądu, nie wyrocznia.

Przykład:

```
2027-01-24␣/etc/letsencrypt/live/zielona.example/cert.pem␣subject=CN=zielona.example
2027-10-03␣/docker/apps/traefik/certy/wildcard-zielona.crt␣subject=CN=*.zielona.example
2046-01-19␣/etc/pki/ca-trust/source/anchors/zielona-root-ca.crt␣subject=C=PL, O=Zielona
sp. z o.o., CN=Zielona-Root-CA
```

□□ **find** = przeszukaj drzewo katalogów według nazwy, typu, daty albo rozmiaru

□□ ZMIENIA SYSTEM

Co naprawdę podaje serwer: openssl s_client -showcerts

```
openssl s_client -connect NazwaSerwera:443 -servername NazwaSerwera -showcerts
</dev/null 2>/dev/null | grep -E '^ *[0-9] s:|^ *i:|NotAfter'
```

Pokazuje każdy certyfikat, który serwer wysyła w połączeniu, z podmiotem (`s:`) i wystawcą (`i:`). Serwer powinien wysłać swój certyfikat i certyfikat urzędu wydającego; brak drugiego wiersza to najczęstsza przyczyna błędu łańcucha na linuksach i w Javie.

Przykład:

```
 0 s:CN=git.zielona.example
   i:C=PL, O=Zielona sp. z o.o., CN=Zielona-CA01
 1 s:C=PL, O=Zielona sp. z o.o., CN=Zielona-CA01
   i:C=PL, O=Zielona sp. z o.o., CN=Zielona-Root-CA
```

□□ **openssl** = zestaw narzędzi do certyfikatów i szyfrowania

□□ TYLKO ODCZYT

Narzędzie pobierania adresów z certyfikatu: certutil -URL

```
certutil -URL NazwaPliku.cer
```

Otwiera okno URL Retrieval Tool, które pobiera każdy adres CDP i AIA z certyfikatu i pokazuje wynik z czasem odpowiedzi. Szybciej niż `certutil -verify -urlfetch` widać tu, który adres nie odpowiada i czy lista jest przeterminowana.

□□ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: okno URL Retrieval Tool z certyfikatem serwera z Zielona-CA01, zaznaczony przycisk CRLs (from CDP), widoczne wiersze z adresem http://pki.ad.zielona.example/pki/Zielona-CA01.crl i statusem Verified.

URL = Uniform Resource Locator: adres strony lub zasobu · **CDP** = CRL Distribution Point: adres w certyfikacie, pod którym urząd publikuje listę unieważnionych certyfikatów · **AIA** = Authority Information Access: miejsce w certyfikacie, które mówi, skąd pobrać certyfikat urzędu, który go podpisał · **certutil** = narzędzie Windows do sprawdzania, importowania i eksportowania certyfikatów z linii poleceń

ZMIENIA SYSTEM

Włączenie dziennika CAPI2: wevtutil sl

```
wevtutil sl Microsoft-Windows-CAPI2/Operational /e:true
```

Dziennik CAPI2 zapisuje każdą próbę budowania łańcucha i sprawdzania unieważnienia w Windows, z dokładnym powodem odrzucenia. Domyślnie jest wyłączony, bo szybko rośnie; włączamy go na czas diagnozy na kliencie albo serwerze NPS, a potem wyłączamy (`/e:false`).

CAPI2 = CryptoAPI 2: warstwa kryptograficzna Windows, która buduje łańcuchy certyfikatów i sprawdza unieważnienia; jej dziennik pokazuje dokładny powód odrzucenia certyfikatu · **NPS** = Network Policy Server: rola Windows Server obsługująca RADIUS, pośredniczy w sprawdzaniu hasła domenowego przy logowaniu do VPN albo Wi-Fi · **wevtutil** = Windows Event Utility: narzędzie wiersza poleceń Windows do eksportu i przeglądania dzienników zdarzeń

OSTROŻNIE · NIEODWRACALNE

Data następnej publikacji listy unieważnień: crl-nextupdate.sh

```
#!/bin/bash
# crl-nextupdate.sh: zwraca czas uniksowy pola nextUpdate listy CRL spod podanego adresu
# Uzycie: crl-nextupdate.sh http://pki.ad.zielona.example/pki/Zielona-CA01.crl
adres="$1"
tmp=$(mktemp)
trap 'rm -f "$tmp"' EXIT
if ! curl -fsS --max-time 10 -o "$tmp" "$adres"; then
  echo 0
  exit 0
fi
```

```
nastepna=$(openssl crl -inform DER -in "$tmp" -noout -nextupdate 2>/dev/null | cut -d= -f2)
if [ -z "$nastepna" ]; then
  nastepna=$(openssl crl -inform PEM -in "$tmp" -noout -nextupdate 2>/dev/null | cut -d= -f2)
fi
if [ -z "$nastepna" ]; then
  echo 0
else
  date -d "$nastepna" +%s
fi
```

Skrypt zapisujemy na `mon01` jako `/usr/local/bin/crl-nextupdate.sh` z prawami `755`. Zwraca `0`, gdy lista jest nieosiągalna albo nieczytelna, więc jeden wyzwalacz łapie oba problemy.

trap = przechwyć sygnał w skrypcie i uruchom w reakcji własny fragment kodu · **echo** = od angielskiego echo: wypisuje podany tekst na standardowe wyjście, często łączone z przekierowaniem do pliku · **exit** = zakończ bieżącą powłokę albo skrypt z podanym kodem wyjścia · **DER** = binarny, nieczytelny dla człowieka format zapisu certyfikatu · **PEM** = Privacy-Enhanced Mail: tekstowy format certyfikatu zaczynający się od -----BEGIN · **date** = wypisz lub ustaw datę i czas, tu do nazw migawek

PLIK KONFIGURACYJNY

Parametr agenta dla list unieważnień: zabbix_agent2.d/pki.conf

```
UserParameter=pki.crl.nextupdate[*],/usr/local/bin/crl-nextupdate.sh "$1"
```

Plik `/etc/zabbix/zabbix_agent2.d/pki.conf` na `mon01`, po zapisaniu `sudo systemctl restart zabbix-agent2`. Pozycje zakładamy na hoście `mon01` z kluczem `pki.crl.nextupdate[http://pki.ad.zielona.example/pki/Zielona-CA01.crl]`, osobno dla każdej listy.

TYLKO ODCZYT

Test parametru i certyfikatu z agenta: zabbix_agent2 -t

```
zabbix_agent2 -t 'pki.crl.nextupdate[http://pki.ad.zielona.example/pki/Zielona-CA01.crl]'
zabbix_agent2 -t 'web.certificate.get[git.zielona.example,443]'
```

Pierwsze polecenie uruchamia nasz skrypt tak jak agent, drugie pobiera certyfikat serwera kluczem wbudowanym we wtyczkę WebCertificate. Wynik drugiego to JSON z polami `x509.not_after.timestamp` i `result.value`, z których korzysta szablon.

Przykład (skrót):

```
pki.crl.nextupdate[http://pki.ad.zielona.example/pki/Zielona-CA01.crl] [s|1800000000]
web.certificate.get[git.zielona.example,443]
[s|{"x509":{"version":3,...,"not_after":{"value":"...","timestamp":...}},"result":{"valu
e":"valid",...}}]
```

JSON = JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami · **zabbix_agent2** = sam proces/binarka agenta Zabbixa drugiej generacji, wywoływana też ręcznie do testów

Z życia wzięte

Z ŻYCIA WZIĘTE

Poniedziałek, 7:35, styczeń. W weekend `dc01` był wyłączony na wymianę zasilacza, a w niedzielę wieczorem wygasła lista unieważnień `CA-Zielona`. Rano NPS na `dc02` odrzuca każdy laptop w Wi-Fi biura, a Kasia nie łączy się z VPN z hotelu. Tomek widzi w dzienniku NPS zdarzenia 6273, a w CAPI2 na `dc02` powód „serwer unieważnień jest w trybie offline”. Po włączeniu `dc01` i `certutil -crl` Wi-Fi wraca w pięć minut, a w Zabbixie od tego dnia jest alarm na trzy dni przed `nextUpdate` każdej listy.

Jak to działa

Rozwiń: jak to działa pod spodem

Spis certyfikatów

Pojęcie: inwentarz certyfikatów. Jedna lista wszystkich certyfikatów, od których coś zależy: gdzie jest, kto go wystawił, do kiedy jest ważny, kto jest właścicielem i jak się go

> odnawia. Bez niej każde wygaśnięcie jest niespodzianką, a każde przenosiny urzędu zgadywaniem.

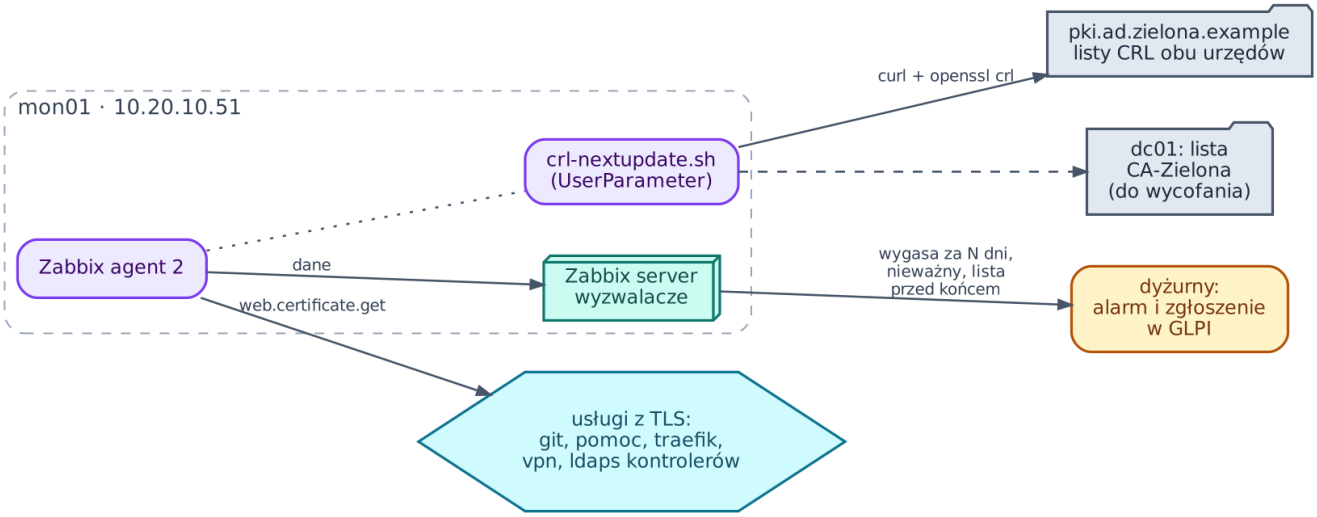
Źródła spisu w Zielonej:

Źródło	Co daje	Jak często
baza urzędu (`certutil -view` na `ca01`, dopóki działa `CA-Zielona`, także na `dc01`)	wszystko, co wydał nasz urząd, z szablonem i wnioskodawcą	raz w miesiącu
magazyny serwerów Windows (`Invoke-Command`)	gdzie certyfikaty są zainstalowane	raz w miesiącu
pliki na linuksach i w `/docker/apps` (skrypt z fiszki)	certyfikaty Let's Encrypt, z `acme01` i ręczne	raz w miesiącu
Zabbix, szablon „Website certificate by Zabbix agent 2”	co naprawdę widzi klient pod każdym adresem	co kilka godzin
dzienniki Certificate Transparency (na przykład wyszukiwarka crt.sh dla `zielona.example`)	certyfikaty publiczne wydane dla naszych nazw, także te, których nie zamawialiśmy	raz w miesiącu
urządzenia: `fw01`, drukarki, przełączniki, BMC serwerów	certyfikaty, których nie widać z Windows ani z linuksów	przy przeglądzie urządzeń

Spis prowadzimy w GLPI (typ zasobu Certyfikaty, powiązany z serwerem i usługą), a nie w arkuszu na udziale. Każdy certyfikat ma właściciela i sposób odnowienia: automatycznie (autorejestracja, ACME, Intune) albo ręcznie z datą przypomnienia 30 dni przed końcem.

Lista miejsc, które ufają naszemu urzędowi poza Windows, jest częścią spisu: `fw01`, linuksy z `update-ca-trust`, kontenery Gitea, Graylog, MailCow i Jenkins z własnym magazynem, Argo CD, telefony z profilem Wi-Fi. Bez niej przenosiny urzędu ze strony Przenosiny urzędu z dc01 bez przerwy: dwa urzędy równolegle i wygaszenie CA-Zielona muszą zgadywać.

Monitoring w Zabbixie



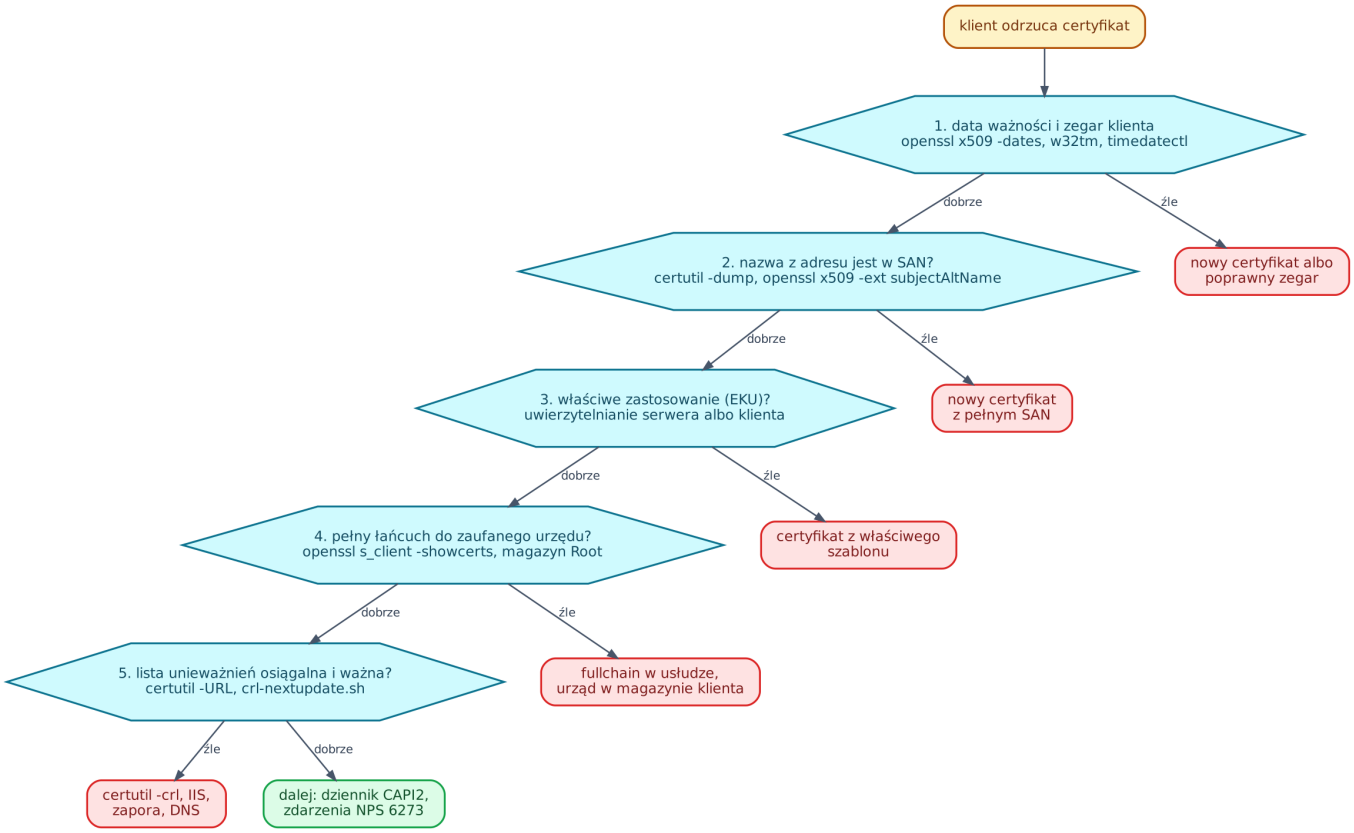
Rysunek 1. Monitoring certyfikatów w Zielonej: agent 2 na mon01 sprawdza web.certificate.get dla każdego adresu z TLS i skryptem crl-nextupdate.sh pobiera listy unieważnień z pki.ad.zielona.example, a Zabbix uruchamia wyzwalacze o wygasającym certyfikacie, nieważnym certyfikacie i liście bliskiej końca (kliknięcie otwiera pełny rozmiar)

Co pilnujemy	Pozycja	Wyzwalacz (propozycja)
certyfikat usługi wewnętrznej	szablon „Website certificate by Zabbix agent 2”, makro `{$CERT.WEBSITE.HOSTNAME}`	domyślnie 7 dni przed końcem; dla certyfikatów rocznych podnosimy `{$CERT.EXPIRY.WARN}` do 30
certyfikat z Let's Encrypt	ten sam szablon	`{$CERT.EXPIRY.WARN}` 14: przy prawidłowym ACME alarm nie powinien się pojawić nigdy
nieważny certyfikat (łańcuch, nazwa)	`$.result.value` z szablonu	wyzwalacz szablonu „Invalid certificate”, waga wysoka
zmiana certyfikatu	odcisk SHA1 z szablonu	informacyjnie; pomaga zauważyć wymianę bez zgłoszenia
lista `Zielona-CA01`	`pki.crl.nextupdate[...Zielona-CA01.crl]`	`last(/mon01/pki.crl.nextupdate[...])-now()<3d` wysoka, wartość `0` katastrofa
lista `Zielona-Root-CA`	`pki.crl.nextupdate[...Zielona-Root-CA.crl]`	30 dni przed końcem: ceremonia
lista `CA-Zielona`	`pki.crl.nextupdate[http://dc01.ad.zielona.example/CertEnroll/CA-Zielona.crl]`	2 dni przed końcem, do wycofania urzędu

Ogólne zasady szablonów i wyzwalaczy opisuje strona Monitorowanie typowych usług. Nazwy pól i makr pochodzą z opisu szablonu na stronie Zabbix dla wersji 7.0 i nowszych; przed wdrożeniem porównujemy je z szablonem zaimportowanym na `mon01`.

Diagnoza w pięciu krokach

Gdy klient odrzuca certyfikat, przyczyna jest prawie zawsze jedna z pięciu: data, nazwa, zastosowanie, łańcuch albo unieważnienie. Sprawdzamy je w tej kolejności, bo pierwsze trzy widać w samym certyfikacie, a dwa ostatnie wymagają sieci.



Rysunek 2. Diagnoza odrzuconego certyfikatu: najpierw data ważności i zegar klienta, potem nazwa w SAN, potem zastosowanie klucza, potem kompletny łańcuch do zaufanego urzędu głównego, na końcu dostępność i ważność listy unieważnień; przy każdym kroku polecenie, które to sprawdza (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: łańcuch certyfikatów.** Ciąg od certyfikatu serwera przez urzędy pośrednie do urzędu głównego. Klient Windows uzupełnia brakujące ogniwa sam, pobierając je spod adresu AIA, ale OpenSSL, Java i większość urządzeń tego nie robi: jeśli serwer nie wyśle certyfikatu urzędu wydającego, połączenie z linuksa się nie uda, a z Windows tak.

Częste błędy

Objaw	Przyczyna	Co robimy
Wi-Fi i VPN odrzucają wszystkich naraz, NPS zapisuje 6273	wygasła lista unieważnień urzędu wydającego (urząd wyłączony dłużej niż zakładka listy)	włączamy urząd, `certutil -crl`, sprawdzamy `nextUpdate`; na przyszłość alarm z fiszki i dłuższa zakładka
`certutil -verify -urlfetch` zgłasza `0x80092013` (serwer unieważnień offline)	adres CDP nieosiągalny z tego komputera (zapora, DNS, IIS)	`certutil -URL`, test `Invoke-WebRequest` pod adres CDP, reguła na `fw01` dla HTTP do `ca01`
Linux i Java: `unable to get local issuer certificate`, Windows bez błędu	serwer nie wysyła certyfikatu urzędu wydającego	w konfiguracji usługi plik z pełnym łańcuchem (`fullchain.pem`), sprawdzenie `openssl s_client -showcerts`
Przeglądarka: `ERR_CERT_COMMON_NAME_INVALID`	nazwa z paska adresu nie występuje w SAN (CN się nie liczy)	nowy certyfikat z wszystkimi nazwami w SAN, także aliasami (`lore` i `lore01`)
VPN: błąd 13801	brak uwierzytelniania serwera w EKU, zły SAN albo brak zaufania do urzędu	strona Błąd 13801: certyfikat bramy VPN, EKU i SAN
`certificate is not yet valid` na jednym serwerze	zegar serwera w przeszłości	`timedatectl` albo `w32tm /query /status`, źródło czasu jak na stronie o kontrolerach
Certyfikat odnowiony, usługa pokazuje stary	usługa nie przeładowała pliku	przeładowanie usługi; przy ACME hak po odnowieniu
Kontener ufa staremu urzędowi, choć serwer ma nowy w `/etc/pki`	obraz ma własny magazyn zaufania	dodajemy urząd do kontenera tak, jak opisuje strona logowania z AD danej aplikacji, i wpisujemy kontener do spisu
Certyfikat działa w biurze, nie działa w hotelu przed VPN	klient sprawdza listę unieważnień, która jest dostępna tylko wewnątrz	decyzja o publicznej publikacji list (strona o projekcie) albo certyfikat bramy z urzędu publicznego
W crt.sh certyfikat dla `zielona.example`, którego nie zamawialiśmy	ktoś przejął walidację domeny (klucz DNS, stary serwer pod naszą nazwą)	zgłoszenie bezpieczeństwa, unieważnienie u urzędu wydającego, wymiana kluczy DNS, rekord CAA z listą dozwolonych urzędów

□□ Ćwiczenia

1. Zabbix zgłasza „Invalid certificate” dla `git.zielona.example`, ale w przeglądarce na Windows wszystko jest w porządku. Od czego zaczynamy?

Odpowiedź

Od `openssl s_client -connect git.zielona.example:443 -servername git.zielona.example -showcerts` z `mon01`. Najczęściej serwer wysyła sam certyfikat bez certyfikatu urzędu wydającego: Windows pobierze brakujące ogniwo spod AIA, a agent na linuksie nie. Poprawka to pełny łańcuch w konfiguracji Traefika albo Gitea.

1. Dlaczego skrypt `crl-nextupdate.sh` zwraca `0` przy błędzie pobrania, zamiast kończyć się błędem?

Odpowiedź

Bo pozycja z błędem staje się w Zabbixie nieobsługiwana i wyzwalacz oparty na wartości milczy. Wartość `0` (rok 1970) jest zawsze „w przeszłości”, więc ten sam wyzwalacz łapie nieosiągalną listę i listę wygasłą.

1. Lista `Zielona-CA01` ma ważność 7 dni i 3 dni zakładki. `ca01` padł w piątek o 16:00, tuż po automatycznej publikacji. Do kiedy mamy czas, zanim NPS zacznie odrzucać laptopy?

Odpowiedź

Lista opublikowana w piątek o 16:00 ma `nextUpdate` dziesięć dni później (7 dni ważności plus 3 dni zakładki), czyli w poniedziałek o 16:00 za półtora tygodnia. Klienci z listą w pamięci podręcznej korzystają z niej do tej daty. Alarm z Zabbixa przyjdzie trzy dni wcześniej.

1. Pan Wojtek prosi o listę wszystkich certyfikatów, które wygasną w pierwszym kwartale 2027 roku. Skąd ją bierzemy?

Odpowiedź

Z GLPI, gdzie spis łączy dane z bazy urzędu, magazynów Windows, plików na linuksach i Zabbixa. Gdy spis nie jest aktualny, robimy go od nowa trzema fiszkami (baza urzędu, `Invoke-Command`, skrypt na linuksach) i filtrujemy po dacie, a do listy dokładamy certyfikaty publiczne z crt.sh.

Powiązane: Monitorowanie typowych usług · Generowanie certyfikatów: OpenSSL i urząd certyfikacji Windows · Certyfikaty SSL: wystawienie, konwersja, weryfikacja · Projekt PKI Zielonej: dlaczego nie na kontrolerze, hierarchia dwupoziomowa, nazwy, ważność, CRL i AIA · Certyfikaty publiczne i ACME: Let's Encrypt z DNS-01, skracanie ważności i step-ca · EAP-TLS i FSSO: logowanie

do VPN certyfikatem użytkownika i grupami z AD

Źródła: (odczyt 20 do 26.10.2026) · zabbix.com/integrations/ssl, „Website certificate by Zabbix agent 2” (makra `{$CERT.WEBSITE.HOSTNAME}`, `{$CERT.EXPIRY.WARN}` z wartością 7, JSONPath `$.x509.not_after.timestamp`, `$.result.value`, wyzwalacze Invalid certificate, Expires soon, Fingerprint changed; wersje 6.0 do 7.4) · zabbix.com, „Zabbix agent 2 items: web.certificate.get” i „User parameters” · learn.microsoft.com, „Certificate provider: -ExpiringInDays” · learn.microsoft.com, „certutil -URL” i „certutil -verify -urlfetch” · learn.microsoft.com, „Troubleshooting PKI problems: CAPI2 diagnostics” · learn.microsoft.com, „NPS event 6273” · docs.openssl.org, `openssl-s_client`, `openssl-crl`, `openssl-x509` · crt.sh (wyszukiwarka dzienników Certificate Transparency)