SICHERHEITS- UND AUTHENTIFIZIERUNGSMUSTER · MODUL 2

Integrierte Authentifizierung mit IIS und Browser

Lektionsleitfaden · Lab- und Prüfungsleitfaden

LearnWisej

Lektionsleitfaden

Die Wahl der IIS-Site-Grenze, die Anbieter der Windows-Authentifizierung und die Reihenfolge der Challenges, die Aushandlung im Browser und die Intranet-Richtlinie, SPNs, Kerberos-Nachweise, Extended Protection und Proxys sowie die Diagnose eines Fehlers nach Schichten. Was das für Entwicklerinnen und Entwickler bedeutet, die die Anmelde-, Identitäts- und Autorisierungspfade einer Wisej.NET-Anwendung im Produktivbetrieb absichern – und wie Sie dieses Modul lesen.

Fokus: Produktivbetrieb

Die Wahl der IIS-Site-Grenze, die Anbieter der Windows-Authentifizierung und die Reihenfolge der Challenges, die Aushandlung im Browser und die Intranet-Richtlinie, SPNs, Kerberos-Nachweise, Extended Protection und Proxys sowie die Diagnose eines Fehlers nach Schichten.

1. Das Problem im Produktivbetrieb

Mitarbeitende öffnen die Intranet-Route von AccessOps und erwarten, mit ihrem Windows-Konto stillschweigend angemeldet zu werden, während Partner eine öffentliche Anmeldeseite erreichen, die nie eine Challenge für Windows-Anmeldedaten stellen darf. Scheitert die stille Anmeldung, ist das Symptom ein Anmeldedialog, eine 401-Schleife oder eine Fehlerseite, und die Ursache liegt in IIS, im Browser, im Verzeichnisdienst oder in einem Proxy, bevor überhaupt Wisej.NET-Code läuft. Dieses Modul zeigt Ihnen, diese Host-Grenze bewusst zu konfigurieren und einen Fehler des Hosts von einer Ablehnung durch die Anwendung zu unterscheiden.

2. Zentrale Konzepte

Konzept	Warum es wichtig ist
Site-Grenze	Eine eigene IIS-Anwendung oder Bindung pro Einstiegspunkt, sodass die reine Windows-Intranet-Route und die anonyme öffentliche Route jeweils genau eine sichtbare Authentifizierungsrichtlinie tragen.
Negotiate-Challenge	Ein HTTP 401 mit `WWW-Authenticate: Negotiate` und/oder `NTLM`; ein berechtigter Browser wiederholt die Anfrage mit einem `Authorization`-Header, und Negotiate wählt Kerberos oder fällt auf NTLM zurück.
Kerberos-Nachweis	Ein Service-Ticket für den kanonischen `HTTP/`-Hostnamen auf dem Client, abgeglichen mit dem Anmeldeeintrag des Servers; der Antwort-Header allein beweist nie, welcher Anbieter gelaufen ist.
Authentifizierung im Kernelmodus	Mit `useKernelMode` führt der Treiber HTTP.sys den Windows-Handshake aus; das Konto, das Tickets entschlüsselt, ist dann normalerweise das Computerkonto, sofern `useAppPoolCredentials` nichts anderes festlegt.
SPN	Der Service Principal Name (`HTTP/accessops.contoso.test`), der der Domäne sagt, für welches Konto ein Ticket verschlüsselt wird; ein fehlender oder doppelter SPN drängt Clients still zu NTLM oder in eine Schleife.
Extended Protection	Prüfungen von Channel Binding und Service Binding, die den Windows-Handshake an die TLS-Verbindung und den erwarteten SPN koppeln – entscheidend, sobald TLS an einem Proxy endet.
Host-Principal in Wisej.NET	Unter ASP.NET Core erreicht der Windows-Benutzer die Anwendung nur als `Application.User`, den von Wisej.NET erfassten `HttpContext.User` der Anfrage; er beweist, wer sich angemeldet hat, nicht, was er lesen darf.

3. Der rote Faden der Lektion

Eine Anwendung pro Einstiegspunkt

AccessOps hat zwei Eingangstüren. Die Intranet-Route ist für Mitarbeitende der Domäne: In IIS installieren Sie den Rollendienst Windows-Authentifizierung, aktivieren ihn für diese Anwendung und deaktivieren dort Anonymous, sodass jede Anfrage eine Challenge erhält, bevor das ASP.NET Core Module sie an die Anwendung übergibt. Die öffentliche Route liefert die Anmeldeseite für Partner und in Modul 5 den OpenID-Connect-Callback. Der Browser des Partners muss diese Seiten erreichen, bevor die Anwendung weiß, wer der Partner ist; diese Anwendung lässt den anonymen Transport deshalb aktiviert, und die Wisej.NET-Anwendung setzt den Zugriff selbst durch. Eine gemeinsame Anwendung würde entweder dem externen Callback eine Challenge stellen oder die Intranet-Bildschirme öffnen.

Halten Sie die Routen in getrennten IIS-Anwendungen oder getrennten Bindungen mit eigenen Anwendungspools. Im Lab ist die Intranet-Route eine isolierte HTTPS-Anwendung mit eigenem Pool, deren Bindung, Ziel-Framework und Authentifizierungszustand Sie zuerst festhalten. Erfassen Sie die übrigen IIS-Modi in einer Bestandsaufnahme, statt sie hinzuzufügen: Basic sendet leicht dekodierbare Anmeldedaten

und braucht überall HTTPS, Digest ist ein veralteter Rollendienst, und die Zuordnung von Clientzertifikaten braucht weiterhin Richtlinien für Vertrauen, Sperrung und Kontozuordnung.

Die Challenge lesen und den Anbieter beweisen

Die erste Anfrage erhält ein 401 mit `WWW-Authenticate: Negotiate` und, falls konfiguriert, `NTLM`. Ein Browser, der den Host für berechtigt hält, wiederholt die Anfrage mit einem `Authorization`-Header, IIS validiert den Austausch, und die Folgeanfrage liefert 200. Die Lab-Fixture F01 hält genau das fest: Eine Negotiate-Challenge gefolgt von 200 ist eine Challenge des Hosts gefolgt von einer Authentifizierung, und nicht mehr. Negotiate ist eine Hülle, die je nach Client, SPN und Anbieterkonfiguration Kerberos transportieren oder auf NTLM zurückfallen kann; ein Negotiate-Header allein beweist also kein Kerberos.

Um Kerberos zu belegen, gleichen Sie zwei Nachweise ab: `klist` auf dem Wegwerf-Client zeigt ein Service-Ticket für `HTTP/accessops.contoso.test`, und der Anmeldeeintrag des Servers für denselben Zeitpunkt nennt Kerberos als Authentifizierungspaket. Fehlt einer davon, notieren Sie den Anbieter als unbestimmt; ein Lab in einer Arbeitsgruppe zeigt Challenges, kann eine Kerberos-Aussage aber nie stützen. Halten Sie Header nur nach Schema und Status fest, denn `Authorization`-Werte sind Anmeldedaten und dürfen weder in einem Trace noch in einem Ticket landen.

Anbieter, Kernelmodus, SPN und der Browser

Die bereichsbezogene Konfiguration des Labs setzt `<anonymousAuthentication enabled="false" />` und `<windowsAuthentication enabled="true" useKernelMode="true">`, wobei die Anbieterliste geleert und als Negotiate, dann NTLM neu aufgebaut wird. Lassen Sie Negotiate an erster Stelle, wenn Sie Kerberos testen, und testen Sie den beabsichtigten Rückfall, statt ihn aus der Reihenfolge abzuleiten. Die Attribute für die Verbindungspersistenz `authPersistNonNTLM` und `authPersistSingleRequest` bestimmen, wie lange einer authentifizierten Verbindung vertraut wird. Laut Pack und der IIS-Dokumentation von Microsoft entschlüsselt der Kernelmodus Tickets normalerweise mit dem Computerkonto; ist der HTTP-SPN für eine eigene Pool-Identität registriert, muss `useAppPoolCredentials` zu dieser Topologie passen. Fragen Sie den SPN mit `setspn -Q` ab und ändern Sie SPNs nur gemeinsam mit dem Verzeichnisteam.

Extended Protection ergänzt `tokenChecking`, `flags` und eine SPN-Liste unter `extendedProtection`; entscheiden Sie zuerst, ob TLS an IIS oder an einem vertrauenswürdigen Zwischenglied endet, und testen Sie dann genau über diesen Pfad. Der Browser hat ein eigenes Mitspracherecht: Edge sendet Anmeldedaten nur stillschweigend an Hosts, die er als Intranet einstuft oder die Administratoren in `AuthServerAllowlist` eintragen; ein Rechner in der Domäne garantiert also keine stille Anmeldung für jeden FQDN, jedes Profil und jeden VPN-Zustand. NTLM und manche Negotiate-Konstellationen brauchen außerdem Verbindungsaffinität, und für den Negotiate-Handler von Kestrel (das Paket `Microsoft.AspNetCore.Authentication.Negotiate`, nicht Teil des Shared Frameworks) verlangt Microsoft hinter einem Proxy eine persistente Verbindungsaffinität im Verhältnis eins zu eins. Ein direkter Test gegen den Server ist Ihre Ausgangsbasis; prüfen Sie den Pfad über den Proxy separat.

Diagnose nach Schichten

Arbeiten Sie sich von außen nach innen vor. Ein 401 vor `Program.Main` gehört zur Challenge des Hosts: Vergleichen Sie, was der Browser gesehen hat, mit den IIS-Logfeldern `sc-status`, `sc-substatus` und `sc-win32-status`, und prüfen Sie dann SPN, Browser-Richtlinie und Proxy. Ein 500.19 direkt nachdem Sie Authentifizierungseinstellungen in die `web.config` geschrieben haben, ist ein Konfigurationsfehler von IIS,

meist ein auf einer übergeordneten Ebene gesperrter Abschnitt: Lesen Sie die gemeldete Datei und den Abschnitt, setzen Sie dieselben Werte in der Location auf Serverebene oder über den IIS-Manager und lassen Sie den Code in Ruhe. Eine Seite, die mit einer gültigen Windows-Identität lädt, aber ein Dokument verweigert, ist eine Ablehnung durch die Anwendung – genau das gewünschte Ergebnis für ein Konto ohne Reader-Grant. Eine Seite, die funktioniert, während eine Dateifreigabe oder ein SQL-Aufruf scheitert, ist der zweite Hop, das Thema von Modul 3.

Sobald IIS die Anfrage authentifiziert hat, erfasst Wisej.NET den `HttpContext.User`, und `Application.User` gibt ihn zurück, sofern die Session keinen eigenen Principal zugewiesen hat. `Application.UserIdentity` gibt es nur im Build für das .NET Framework. Der Start der AccessOps-Intranet-Route behandelt eine fehlende Windows-Identität deshalb als Fehler des Hosts und schlägt den Grant separat nach:

```
static void Main()
{
    // IIS hat die Challenge beantwortet; Wisej hat HttpContext.User erfasst.
    if (!Application.IsAuthenticated ||
        Application.User.Identity is not WindowsIdentity windows)
    {
        new HostFaultPage().Show();   // Host-Schicht, keine Ablehnung der App
        return;
    }

    // Das Windows-Konto ist ein Nachweis; der Grant ist die Berechtigung.
    var grant = AccessGrants.FindForAccount(windows.User?.Value);
    new DocumentsPage(grant).Show(); // DocumentService lehnt einen null-Grant ab
}
```

Das Lab beweist diese Trennung zweimal: Unter IIS in einer Windows-Domäne wird ein zweites Konto ohne Reader nach erfolgreicher Authentifizierung durch den Host abgelehnt; auf jedem Entwicklerrechner lehnt das PolicyHarness eine authentifizierte Identität ab, deren Grant nicht lesen darf – bei null Repository-Lesezugriffen. Sichern Sie die ursprünglichen IIS-Einstellungen, damit das Rollback ein einziger Schritt ist.

Sicherheitsregel

Halten Sie die Identitäten auseinander und entscheiden Sie auf dem Server. Der Host-Principal, der Wisej.NET-Anwendungs-Principal und das effektive Windows-Token sind drei verschiedene Fakten, und keines davon ist eine Berechtigung. Jeder geschützte Lesezugriff prüft den aktuellen Grant, den Mandanten und den Datensatz in der Service-Schicht, bevor das Repository läuft, und ein ausgeblendeter Button, ein Rollen-Claim oder ein abgemeldeter Tab beweist nichts, solange diese Prüfung nicht ablehnt.

Lernziele

Am Ende dieses Moduls können Sie:

- ✓ Den IIS-Authentifizierungsmodus für jeden Einstiegspunkt von AccessOps wählen – reines Windows im Intranet, anonymer Transport für die öffentliche Anmeldung und den Callback – und

beide in getrennten Anwendungen oder Bindungen halten.

- ✓ Eine 401-Challenge mit Negotiate oder NTLM lesen und anhand abgeglichener Nachweise aus Ticket und Server-Anmeldung statt des Antwort-Headers belegen, ob Kerberos verwendet wurde.
- ✓ Anbieter der Windows-Authentifizierung, den Kernelmodus, den HTTP-SPN und Extended Protection konfigurieren und erklären, wie die Intranet-Richtlinie des Browsers und die Verbindungsaffinität eines Proxys die stille Anmeldung beeinflussen.
- ✓ Einen Fehler nach Schichten diagnostizieren: Challenge des Hosts, Konfigurationssperre in IIS (500.19), Ablehnung durch die Anwendung und Zugriff über den zweiten Hop.

Lab- und Prüfungsleitfaden

Was Sie im praktischen Lab bauen, das vorgeschlagene Vorgehen und die geforderten Ergebnisse.

Praktisches Lab

Veröffentlichen Sie die Intranet-Route von AccessOps in eine isolierte HTTPS-Anwendung in IIS mit eigenem Anwendungspool, halten Sie Bindung, Ziel-Framework und Authentifizierungszustand fest und wenden Sie dann die bereichsbezogenen Location-Einstellungen für windows-auth an, um die Windows-Authentifizierung zu aktivieren und Anonymous zu deaktivieren. Öffnen Sie von einem Client in der Domäne den kanonischen FQDN als CONTOSO\reader1, halten Sie das Challenge-Schema, das Ergebnis der Folgeanfrage und die geschwärzte Anwendungsidentität fest, fragen Sie den HTTP-SPN ab und vergleichen Sie das Ticket des Clients mit den Anmeldenachweisen des Servers; melden Sie sich dann mit einem zweiten Konto ohne Reader an und erfassen Sie eine Ablehnung durch die Anwendung nach erfolgreicher Authentifizierung durch den Host, bevor Sie die ursprünglichen Einstellungen wiederherstellen. Die Challenge, der ausgehandelte Anbieter und die SPN-Nachweise lassen sich nur in der Lab-Umgebung mit IIS und einer Windows-Domäne prüfen (eine Arbeitsgruppe zeigt Challenges, kann Kerberos aber nicht belegen – halten Sie dort ein unbestimmtes Ergebnis fest); auf einem einfachen Entwicklerrechner beweist das PolicyHarness lokal, dass eine authentifizierte Identität ohne Reader-Grant trotzdem abgelehnt wird, bei null Repository-Lesezugriffen.

Ausgangspunkt

Öffnen Sie die AccessOps-Projektmappe aus Modul 1 mit ihrem `DocumentService`, dem lesezählenden Repository, den Fixtures für die Mandanten A und B und dem eigenständigen PolicyHarness. Sie brauchen eine isolierte HTTPS-Anwendung in IIS, zwei Testkonten in der Domäne, Edge und die Berechtigung, die IIS-Logs zu lesen; ohne Domäne führen Sie die IIS-Schritte in einer Arbeitsgruppe und die Richtlinienschritte im PolicyHarness aus.

Vorgeschlagenes Vorgehen

- ✓ Veröffentlichen Sie die Intranet-Route in eine eigene IIS-Anwendung mit eigenem Pool und sichern Sie die aktuellen Authentifizierungseinstellungen, die Bindung und das Ziel-Framework als Ausgangsbasis für das Rollback, bevor Sie etwas ändern.
- ✓ Wenden Sie die bereichsbezogenen Location-Einstellungen an (Anonymous aus, Windows an, Kernelmodus, Anbieter geleert und als Negotiate, dann NTLM neu aufgebaut); antwortet IIS mit 500.19, setzen Sie dieselben Werte auf Serverebene oder im IIS-Manager, statt Abschnitte zu entsperren.
- ✓ Öffnen Sie `https://accessops.contoso.test` von einem Client in der Domäne als `CONTOSO\reader1` und halten Sie das 401-Schema, den Status der Folgeanfrage und die geschwärzte Identität aus der Diagnoseansicht fest, nie den Wert von `Authorization`.
- ✓ Führen Sie `setspn -Q HTTP/accessops.contoso.test` und `klist` aus, gleichen Sie das Ticket des Clients mit dem Anmeldeeintrag des Servers ab und notieren Sie allein anhand dieser Nachweise Kerberos, NTLM oder unbestimmt.
- ✓ Melden Sie sich mit dem zweiten Konto ohne Reader an, erfassen Sie die Ablehnung durch `DocumentService` bei null Repository-Lesezugriffen, ergänzen Sie den passenden Fall im

PolicyHarness, stellen Sie dann die gesicherten Einstellungen wieder her und bestätigen Sie, dass die Site funktioniert.

Geforderte Ergebnisse

- ✓ Wirksame IIS-Authentifizierungseinstellungen für die Intranet- und die öffentliche Route
- ✓ Geschwärzter Challenge-Trace mit Schema, Status und Ergebnis der Folgeanfrage
- ✓ SPN- und Ticket-Nachweise oder ein ausdrückliches Ergebnis „Anbieter unbestimmt“
- ✓ Vom Host authentifizierter Benutzer, der von der AccessOps-Richtlinie abgelehnt wird, mit einem PolicyHarness-Fall für dieselbe Regel
- ✓ Diagnosenotiz nach Schichten, die einen IIS-Fehler von einer Ablehnung durch die Anwendung trennt, mit den Einstellungen für das Rollback

Wiederholungsfragen

- ✓ Der Browser zeigt eine Negotiate-Challenge gefolgt von 200, und die Diagnoseansicht zeigt den richtigen Kontonamen. Was können Sie über Kerberos aussagen, und welcher Nachweis fehlt noch?
- ✓ Ein Partner meldet einen Windows-Anmeldedialog auf der öffentlichen Anmeldeseite. Welche IIS-Einstellung oder Grenze würden Sie zuerst prüfen, und warum liegt die Korrektur nicht im Wisej.NET-Code?
- ✓ Die stille Anmeldung funktioniert, wenn Sie den Server direkt aufrufen, läuft über den Load Balancer aber in eine Schleife. Welche zwei Eigenschaften des Proxy-Pfads würden Sie vergleichen, und was würde jede davon kaputtmachen?

Sicherheitsregel

Halten Sie die Identitäten auseinander und entscheiden Sie auf dem Server. Der Host-Principal, der Wisej.NET-Anwendungs-Principal und das effektive Windows-Token sind drei verschiedene Fakten, und keines davon ist eine Berechtigung. Jeder geschützte Lesezugriff prüft den aktuellen Grant, den Mandanten und den Datensatz in der Service-Schicht, bevor das Repository läuft, und ein ausgeblendeter Button, ein Rollen-Claim oder ein abgemeldeter Tab beweist nichts, solange diese Prüfung nicht ablehnt.