# 36. Model i technologie zapewnienia bezpieczeństwa w środowisku usług webowych

Problem bezpieczeństwa w środowisku usług webowych wynika przede wszystkim z jego rozproszonego charakteru i jest związany przede wszystkim z zagrożeniami dla systemów udostępniających usługi, dla aplikacji klienckich z nich korzystających oraz dla całej infrastruktury sieciowej w ramach której prowadzona jest komunikacja.

Niebezpieczeństwa związane z komunikacją są istotne również, ze względu na formę komunikacji. Komunikacja pomiędzy usługami webowymi odbywa się z wykorzystaniem protokołu SOAP, który jako nośnik informacji wykorzystuje struktury danych zapisane w języku XML, a nie, np. zapis binarny danych. Oznacza to, że w przypadku przechwycenia takich komunikatów w bardzo prosty sposób można odczytać z nich dane lub je zmodyfikować.

Ze względu na to zapewnienie bezpieczeństwa w środowiskach usługowych jest bardzo istotne. W ramach tego środowiska możemy wyróżnić dwie główne warstwy w ramach których może, i powinno, zostać zapewnione bezpieczeństwo:

***Warstwę Transportową*** – warstwa odpowiedzialna za przesyłanie danych poprzez fizyczną infrastrukturę komunikacyjną pomiędzy dwoma punktami

***Warstwę Wiadomości*** – warstwa odpowiedzialna za przesyłanie wiadomości za pomocą protokołu SOAP.

Zapewnienie bezpieczeństwa w warstwie transportowej wiąże się przede wszystkim z wykorzystaniem już istniejących rozwiązań w dziedzinie bezpieczeństwa sieciowego takich jak bezpieczne połączenie SSL lub TLS.

Zastosowanie mechanizmów bezpieczeństwa tylko w tej warstwie nie gwarantuje jednak bezpieczeństwa z punktu widzenia architektury SOA. Ma to związek z tym, że tego typu zabezpieczenia gwarantują bezpieczeństwo w trakcie transmisji (kolokwialnie mówiąc: „gdy dane są w kablu”). Oznacza to, że jeżeli w komunikacji biorą udział węzły pośredniczące, co ma miejsce bardzo często w architekturze SOA (np. w postaci Brokera usług lub silnika wykonawczego usług złożonych), wiadomość nie jest zabezpieczona, gdy jest przez nie przetwarzana i przesyłana dalej. Łatwo sobie wyobrazić sytuację, w której taki element infrastruktury padł ofiarą ataku i działa na nim oprogramowanie szpiegujące odbierane i wysyłane komunikaty.

Aby wyeliminować tę lukę bezpieczeństwa konieczne jest zastosowanie mechanizmów bezpieczeństwa w warstwie wiadomości. Zapewnienie bezpieczeństwa w warstwie wiadomości wiąże się przede wszystkim ze wzbogaceniem protokołu wymiany wiadomości – SOAP – o mechanizmy pozwalające na zagwarantowanie wiadomościom, m. In. poufności i integralności.

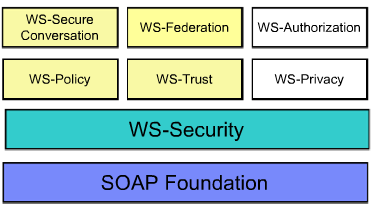
Opis architektury usług webowych zaproponowany przez W3c z lutego 2004r. wprowadza trzy główne pojęcia związane z bezpieczeństwem usług webowych w warstwie wiadomości:

* Zasób (ang. Resource) – są to dane, które mają być zabezpieczone
* Polityka (ang. Policy) – są to ograniczenia/wymagania nakładane na aktorów biorących udział w komunikacji. Wyróżniane są dwa typy polityk: Permission Policies , które opisują jakie akcje mogą zostać wykonane przez poszczególnych aktorów oraz Obligatory Policies, które definiują jakie akcje muszą zostać wykonane, np. w celu uzyskania dostępu i niezbędnych zezwoleń.
* Mechanizmy zabezpieczające zasoby (nazywanie również Strażnikami Polityk, ang. Policy Guards) – mechanizmy te mają na celu zabezpieczenie zasobów oraz dbają o to, aby zdefiniowane polityki były realizowane

W dalszej części opis architektury definiuje zagrożenia dla architektury występujące w warstwie wiadomości takie jak *zagrożenie integralności wiadomości*, *zagrożenie poufność*, *ataki Man-in-the-middle*, *ataki DDOS*, itp. Opisane zostały również wymagania dla architektury usługowej z punktu widzenia tych zagrożeń, takie jak konieczność istnienia mechanizmów autentykacji, autoryzacji, mechanizmów zapewniających integralność oraz poufność, itp.

Jednakże, pomimo wprowadzenia trzech podstawowych pojęć związanych z bezpieczeństwem w ramach architektury usług webowych oraz pomimo identyfikacji zagrożeń bezpieczeństwa oraz mechanizmów, które powinna implementować architektura w celu zapewnienia bezpieczeństwa, standard ten w żaden sposób nie opisuje konkretnych metod zapewnienia bezpieczeństwa, ani nie definiuje żadnych formalnych standardów mających na celu zapewnienia bezpieczeństwa infrastrukturze.

Kwestie bezpieczeństwa w architekturze zorientowanej na usługi został rozwinięty dopiero przez grupę OASIS przy współpracy z IBM, BEA System, Microsoft oraz innymi. W ramach prac prowadzonych przez tę grupę zaproponowane rozszerzenie podstawowego protokołu SOAP o elementy mające na celu gwarantowanie bezpieczeństwa w środowisku usług webowych, które weszły w skład proponowanego przez nich modelu referencyjnego SOA, który jest podstawą do wszystkich obecnie stosowanych implementacji tej architektury. Jako główne rozszerzenie zaproponowane przez OASIS można wskazać Standard *WS-Security*, na bazie którego zostały nadbudowane kolejne standardy takie jak *WS-Policy*, *WS-Trust*, *WS-Privacy, WS-SecureConversation*, *WS-Federation* oraz *WS-Authorization*.



## WS-Security

WS-Security można uznać za nadbudowany na protokole SOAP protokół komunikacyjny zapewniający środki do zastosowania mechanizmów bezpieczeństwa podczas wymiany wiadomości w systemach usługowych. Zapewnia on podstawy do wykorzystania mechanizmów autentykacji i autoryzacji oraz zapewnia wspierające go specyfikacje, które razem tworzą swego rodzaju Framework do tworzenia i wdrażania strategii bezpieczeństwa w architekturze SOA. Głównym celem przyświecającym stworzeniu WS-Security było zapewnienie czegoś więcej niż bezpieczeństwa w warstwie transportowej. WS-Security skupia się na dostarczeniu modelu dla wiadomości SOAP, który pozwoliłbym na zapewnienie pełnego bezpieczeństwa od samego początku życia wiadomości do samego końca, gdy dotrze ona do usługi docelowej przechodząc w międzyczasie przez wielu pośredników (tzw. zapewnienie bezpieczeństwa end-to-end).

W celu zapewnienia bezpieczeństwa wiadomości w trakcie przesyłania wiadomości pomiędzy poszczególnymi pośrednikami WS-Security implementuje mechanizmy bezpieczeństwa poprzez wykorzystanie nagłówka (element „header”) wiadomości SOAP (dane wymieniane pomiędzy usługami i klientem znajdują się w ciele, elemencie „body” wiadomości). WS-Security pozwala na zaimplementowanie dowolnego mechanizmu autentykacji w postaci tokenów, jednakże jego specyfikacja skupia się na wykorzystaniu tokenów trzech głównych postaciach:

* Token w postaci nazwy użytkownika i hasła
* Token w postaci Certyfikatu X.509
* Token w postaci Kerberos

Oprócz mechanizmów autentykacji opartych o tokeny WS-Security oferuje również możliwość podpisywania wiadomości, aby możliwe było potwierdzenie jej integralności i braku modyfikacji ze strony osób trzecich w momencie jej odebrania.

Jeżeli zapewnienie integralności nie jest wystarczające i wymagana jest również poufność WS-Security umożliwia również zaszyfrowanie sekcji „body” w wiadomości SOAP lub jej części z wykorzystaniem wybranego algorytmu do szyfrowania.

Na bazie WS-Security zostały nadbudowane inne specyfikacje rozszerzające jego możliwości . Bazową warstwę rozszerzeń stanowią specyfikacje związane z politykami, takie jak WS-Policy, WS-Trust i WS-Privacy. Z kolei na ich bazie zbudowane zostały kolejne specyfikacje takie jak WS-SecureConversation, WS-Federation oraz WS-Authorization.

## WS-Policy

Element ten pozwala na opisanie polityk powiązanych z poszczególnymi elementami zdefiniowanymi w ramach WS-Security (warto jednak zaznaczyć, że pozwala on na definiowanie dowolnych polityk, nie tylko bezpieczeństwa). Zdefiniowane polityki bezpieczeństwa mogą być wyrażone za pomocą WS-Policy w postaci asercji, które mogą być zastosowane do pojedynczych usług lub i group.

Specyfikacja WS-Policy opisuje między innymi to w jaki sposób nadawcy oraz odbiorcy mogą specyfikować swoje wymagania i możliwości. Specyfikacja ta jest w pełni rozszerzalna i nie nakłada żadnych ograniczeń na metody opisu polityk, jednak definiuje ona kilka podstawowych atrybutów takich jak atrybuty prywatności (privacy attributes), formaty kodowania (encoding formats), wymagania na tokeny bezpieczeństwa (security token requirements) oraz wspierane algorytmy.

## **WS-Trust**

WS-Trust opisuje standardowy model, który jest wykorzystywany do unifikowania istniejących modeli zaufania. WS-Trust pozwala zapewnia między innymi mechanizm komunikacyjny pozwalający na włączenie w proces autentykacji zewnętrznych dostawców usług autentykacji, w celu potwierdzenia i weryfikacji aktorów biorących udział w komunikacji.

Specyfikacja opisuje w jaki sposób istniejące bezpośrednie relacje zaufania pomiędzy aktorami mogą zostać wykorzystane jako podstawa do pośrednictwa informacji o zaufanych aktorach poprzez tworzenie i wydawanie tokenów zaufania oraz to w jaki sposób te tokeny mogą zostać wykorzystane w celu zapewnienia bezpieczeństwa.

## **WS-Privacy**

WS-Privacy może być wykorzystywane do informowania o politykach ochrony prywatności związanych z oferowaną usługą oraz do sprawdzenia czy osoba lub system jej żądający spełnia te polityki. Działanie WS-Privacy jest ściśle powiązane z WS-Policy oraz WS-Trust. Poprzez wykorzystanie WS-Policy, WS-Security oraz WS-Trust organizacje mogą określać i wskazywać zgodności z ustalonymi zasadami ochrony.

Specyfikacja ta opisuje model pozwalający na zdefiniowanie polityk ochrony prywatności wewnątrz opisów WS-Policy. Dodatkowo opisuje sposób wykorzystania WS-Security w celu powiązania poświadczeń prywatności z poszczególnymi wiadomościami. Ostatecznie, specyfikacja opisuje w jaki sposób mechanizmy WS-Trust mogą zostać wykorzystane do oceny tych poświadczeń.

## WS-SecureConversation

Specyfikacja WS-SecureConversation opisuje w jaki sposób usługa webowa może uwierzytelniać aktorów przesyłających komunikaty oraz jak nawiązywać wzajemnie uwierzytelnioną komunikację. W celu zapewnienia narzędzi do tego specyfikacja opisuje w jaki sposób ustalać klucze sesji oraz klucze dla każdej wiadomości z osobna. Poza tym, specyfikacja ta opisuje w jaki sposób usługi mogą bezpiecznie wymieniać kontekst bezpieczeństwa (zbiory poświadczeń na temat atrybuty bezpieczeństwa i powiązanych z nimi). Aby to osiągnąć, specyfikacja opisuje i opiera się na koncepcji wydawania tokenów bezpieczeństwa i na ich wymianie na podstawie mechanizmów zapewnianych przez WS-Security i WS-Trust.

## **WS-Federation**

Istnieje wiele sposobów na integracje różnych domen zaufania oraz różnych metod poświadczeń wykorzystywanych w standardach WS-Security, WS-Policy oraz WS-Trust. WS-Federation ma na celu dostarczenie szeregu standardów oraz modelu bezpieczeństwa, który pozwoli na utworzenie federacji w środowisku, w którym poziom zaufania został ustalony pomiędzy różnymi domenami zaufania. Jako przykład można podać sytuację, w której część infrastruktury usługowej, którą możemy uznać za domenę zaufania wykorzystuje do autentykacji protokół Kerberos, a część wykorzystuje do autentykacji certyfikaty X.509. WS-Federation ma na celu umożliwienie ustalenia pewnego wspólnego poziomu zaufania pomiędzy tymi domenami oraz umożliwienie nawiązanie komunikacji pomiędzy elementami należącymi do obu tych grup zaufania.

## **WS-Authorization**

Specyfikacja ta opisuje w jaki sposób są określone i zarządzane polityki dostępu do usług webowych. W szczególności opisuje ona w jaki sposób mają być specyfikowane poświadczenia w ramach tokenów bezpieczeństwa oraz w jaki sposób te poświadczenia mają być interpretowane przez punkt końcowy komunikacji, czyli np. usługę webową. Specyfikacja ta została zaprojektowana przede wszystkim z myślą o zapewnieniu dużej elastyczności i rozszerzalności zarówno w przypadku formatu autoryzacji jak i języka autoryzacji. Decyzja ta miała na celu umożliwienie zastosowania tej specyfikacji w ramach szerokiej gamy scenariuszy oraz zapewnienie długiej żywotności temu rozwiązaniu.