Z powyższego ćwiczenia można wyciągnąć kilka wniosków dotyczących zarządzania i zabezpieczania oprogramowania za pomocą funkcji AppLocker w PowerShell:

1. **Planowanie**: Przed wdrożeniem funkcji AppLocker należy dokładnie przemyśleć, jakie cele chcemy osiągnąć i jakie reguły będą najlepiej odpowiadać naszym potrzebom. Decyzja między czarną listą a białą listą oraz wybór kryteriów (ścieżka, wydawca, skrót) są kluczowe dla skuteczności zabezpieczeń.

2. **Praca detektywa**: Konieczne jest zidentyfikowanie plików wykonywalnych, na które chcemy nałożyć reguły, czy to poprzez ścieżki, wydawców czy skróty. Przygotowanie listy plików i ich analiza jest niezbędna przed wdrożeniem zasad AppLocker.

3. **Wykorzystanie poleceń PowerShell**: Dzięki poleceniom cmdlet w PowerShell związanych z funkcją AppLocker, można łatwo zarządzać zasadami, tworzyć nowe polityki, testować je i aplikować. Jest to efektywne narzędzie do automatyzacji procesów związanych z zarządzaniem bezpieczeństwem oprogramowania.

4. **Testowanie i diagnostyka**: Przed wdrożeniem reguł AppLocker zalecane jest przeprowadzenie testów, aby upewnić się, że zasady nie blokują niepożądanych plików ani aplikacji. Testowanie pozwala również na diagnozowanie ewentualnych problemów z zastosowanymi zasadami.

5. **Importowanie zasad**: Generowanie plików XML z zasadami AppLocker nie jest **równoznaczne** z ich zastosowaniem. Konieczne jest importowanie tych zasad do lokalnego zestawu reguł lub do obiektu zasad grupy, aby faktycznie zabezpieczyć system.

Podsumowując, skuteczne zarządzanie funkcją AppLocker za pomocą PowerShell wymaga starannego planowania, identyfikacji plików, wykorzystania odpowiednich poleceń cmdlet oraz testowania i aplikowania zasad w sposób odpowiedni do potrzeb i środowiska organizacji.