Załącznik nr 2 do OPZ

Wytyczne Departamentu Bezpieczeństwa dla testów penetracyjnych systemów CeZ

Bezpieczeństwo systemów i aplikacji

METRYKA DOKUMENTU

OPIS	Wytyczne Departamentu Bezpieczeństwa dla testów penetracyjnych dla systemów CeZ
WŁAŚCICIEL	Dyrektor Departamentu Bezpieczeństwa
ZAŁĄCZNIKI	Wymagania ASVS.xls
WERSJA	1.0
KLASYFIKACJA	Do użytku publicznego.

1. Cel dokumentu

Systemy CeZ są systemami wysokiego ryzyka, niektóre z nich są wykorzystywane do świadczenia usługi kluczowej w Polsce. Celem dokumentu jest wskazanie wytycznych dla testów bezpieczeństwa - penetracyjnych – systemów i aplikacji rozwijanych i utrzymywanych przez CeZ, głównie w oparciu o OWASP Application Security Verification Standard v4.0.3 (ASVS).

1. Zakres testów

1.1. Standard bezpieczeństwa OWASP ASVS

OWASP Application Security Verification Standard (ASVS) jest listą wymagań służącą do definiowania, budowania, testowania i weryfikacji bezpiecznych aplikacji. OWASP ASVS definiuje trzy poziomy weryfikacji bezpieczeństwa, z których każdy jest coraz głębszy.

- Poziom L1 (128 wymagań) jest przeznaczony dla aplikacji które nie przetwarzają poufnych informacji.
- Poziom L2 (259 wymagań) jest przeznaczony dla aplikacji przetwarzających dane osobowe i transakcyjne i jest zalecanym poziomem dla większości aplikacji.
- Poziom L3 (278 wymagań) jest przeznaczony dla najbardziej krytycznych aplikacji, które wykonują transakcje o wysokiej wartości, zawierają wrażliwe dane medyczne lub jakiekolwiek aplikacje, które wymagają najwyższego poziomu zaufania.

Dokument OWASP ASVS jest dostępny tutaj: https://github.com/OWASP/ASVS na licencji Creative Commons Attribution ShareAlike, tłumaczenie wymagań na polski wykonano w CeZ.

Zgodność aplikacji EWP z wymaganiami ASVS poziomu L1 zostanie zweryfikowana przez skanowanie podatności oraz manualne testy penetracyjne. Ocena zostanie przeprowadzona zgodnie techniką black-box, wedle założeń której zespół testujący przystępując do realizacji testów ma wiedzę o przedmiocie testów na poziomie analogicznym jak inni jej użytkownicy.

1.2. Zakres testów penetracyjnych

1.2.1. W ramach testów penetracyjnych **należy** sprawdzić wszystkie wymagania L1, należy **pominąć** wymagania dotyczące technologii które nie zostały wykorzystane w testowanej aplikacji, wskazując w raporcie z testów, które wymagania zostały pominięte.

1.2.2. W ramach testów penetracyjnych **można** sprawdzić wymagania L2, o ile dotyczą testowanej aplikacji i zostanie to uzgodnione przed zleceniem testów.

1.2.3. Załącznik 1 - wzorcowa lista kontrolna (‘Wymagania ASVS.xlsx’) wskazuje opis i polskie rozumienie wymagań ASVS, które powinny zostać zweryfikowane w ramach wykonywanych testów penetracyjnych.

1.2.4. Testy mają być prowadzone zgodnie z OWASP Web Security Testing Guide.

1.2.5. W przypadku, gdy testowany system wykorzystuje konteneryzację, należy zweryfikować pod kątem bezpieczeństwa:

1.2.5.1. Kubernetes – należy wykonać audyt poprawności pod kątem bezpieczeństwa konfiguracji Kubernetes, co najmniej zgodnie z dokumentacją Kubernetes – Security, w tym Security Checklist oraz OWASP Kubernetes Security Cheat Sheet. Weryfikacja możliwości nadużycia konfiguracji Klastra powinna uwzględnić co najmniej:

1.2.5.1.1. Sprawdzenie konfiguracji RBAC (Role-Based Access Control) polegającą na wyszukaniu kont mających podwyższone uprawnienia pozwalające np. na: listowanie sekretów, tworzenie PODów.

1.2.5.1.2. Weryfikację dostępów do „Resource” oraz „Verb”.

1.2.5.1.3. Sprawdzenie wyników enumeracji subdomen w celu wykrycia nadmiarowo upublicznionych PODów.
1.2.5.1.4. Sprawdzenie otwartych portów.
1.2.5.1.5. Sprawdzenie sieciowych pluginów.
1.2.5.1.6. Weryfikację możliwości uzyskania anonimowego dostępu do API oraz komponentu ETCD.
1.2.5.1.7. Weryfikację czy może dojść upublicznienia nazw PODów poprzez port read-only dla „Kubelet".
1.2.5.1.8. Weryfikację uprawnienia do użycia „Pods/Exec".
1.2.5.1.9. Weryfikację uprawnienia do użycia „Get/Patch" dla przypisywania ról.
1.2.5.1.10. Weryfikacja uprawnień do podszywania się pod inne role – funkcja „Impersonate".
1.2.5.1.11. Sprawdzenie możliwości pobrania tokenów.
1.2.5.1.12. Weryfikację pod kątem możliwości wyjścia z PODa do węzła.
1.2.5.1.13. Pod kątem poprawnego rozdzielenia DMZ i części bezpiecznej,
1.2.5.1.14. Możliwości eskalacji uprawnień,
1.2.5.1.15. Nadmiernego używania konfiguracji domyślnych Kubernetes, co najmniej pod kątem uniknięcia pomyłek konfiguracyjnych.

1.2.5.2. Docker – audyt poprawności pod kątem bezpieczeństwa konfiguracji Docker, co namniej zgodnie z dokumentacją Docker – Security oraz OWASP Docker Security Cheat Sheet.

2. Warunki prowadzenia testów

2.1. Testowany system w zależności od rodzaju zlecenia na testy należy traktować jako
2.1.1. Whitebox – korzystamy z dostępu do testowanego systemu, systemu operacyjnego i kodu źródłowego lub
2.1.2. Greybox – nie korzystamy dostępu do testowanego systemu, systemu operacyjnego i kodu źródłowego.
2.2. Spotkanie techniczne
2.2.1. Zamawiający wymaga, aby Strony przed przystąpieniem do realizacji testów wzięły udział w tzw. spotkaniu technicznym w celu szczegółowego określenia metodyki pracy i zasad współpracy w zakresie przeprowadzenia przedmiotowych testów.
2.2.2. Zamawiający wymaga, aby ze strony Wykonawcy w spotkaniu wzięły udział osoby wskazane w liście testerów skierowanych do realizacji testów (Wykaz osób stanowiący Załącznik nr 3 do Umowy).
2.2.3. Spotkanie może odbyć się w siedzibie Zamawiającego lub poprzez wideokonferencję w terminie ustalonym przez Stron.
2.3. Należy logować czas wykonania testów: start data+godz., stop data+godz – o realne czasy wykonania testu, np. przy testach manualnych pomijamy noc, jeśli w nocy nikt nie testuje systemu.
2.4. Należy zachowywać dowody prowadzonych testów – kompletne pliki logów testowanego systemu oraz narzędzi testujących, zrzuty ekranu etc.
2.5. Testy bezpieczeństwa powinny wykonywać osoby mające do tego kompetencje, które Wykonawca może udowodnić.
2.6. Należy zapewnić możliwość analizy statycznej kodu w celu spełnienia wymagań ASVS L1.

3. Wymagania dotyczące raportu z testów

3.1. Ocena poziomu bezpieczeństwa powinna znaleźć się w podsumowaniu w nagłówku dokumentu – streszczenie oceny (3 zdania) wraz z tabelką podsumowania.
3.2. Opis zakresu testów, powinien zawierać co najmniej:
3.2.1. Opis testów (co testujemy) oraz standardy o jakie się opierają się testy.
3.2.2. Oznaczenie systemu, modułów systemu (jeśli można je wyróżnić), wersje systemu i modułów.
3.2.3. Oznaczenie środowiska używanego do testów.
3.2.4. Narzędzia wykorzystywane podczas testów. Jeśli była dołączona własna konfiguracja narzędzia, powinna być informacja o tym, można pominąć szczegóły konfiguracji.

3.2.5. Informacja, czy testy zostały wykonane manualnie, czy automatycznie.
3.2.6. Czas wykonania testów, których dotyczy raport.
3.2.7. Osoby biorące udział w testach, w jakiej roli, zakres odpowiedzialności za testy – opisowo, np. „wykonanie testów automatycznych", „testy modułu X", „testy w obszarze Kubernetes".
3.2.8. Testowana wersja każdego systemu/modułu tego systemu, informacja jeśli się zmieniła w trakcie testów tak, żeby można było określić co przetestowano na nowszej a co na starszej wersji, w tym informacja o dacie i godz. zmiany (przybliżona, jeśli nie znana dokładna).
3.2.9. Zmiany w trakcie testów – wynikające np. z poprawek lub przekazania nowych f-ści i ocena czy to wpłynęło na testy (czemu tak, czemu nie, czy ponowiono testy po zmianie, w jakim zakresie)
3.2.10. Jakie serwery wirtualne brały udział w teście? – nazwa, IP, rola serwera (DB/APP/inne), komponent(y) systemu na danym serwerze
3.2.11. W jaki sposób testowano – próby wejścia od zewnątrz, od wewnątrz?, jakie urządzenia były między hostem testującym, a testowanym - każdy host oddzielnie czy przez load balancer; Jeśli jeden host lub przez LB, to trzeba jasno napisać, że na wszystkich hostach jest to samo oprogramowanie w tej samej wersji.
3.2.12. Dowody wykonanych testów jako załączniki – logi testowanej aplikacji, logi narzędzi testowych, eksport wyników z tych narzędzi w postaci nieprzetworzonej, zrzuty z narzędzi monitorujących.
3.2.13. Tabela z informacją które ww załączniki są wynikiem wykonania jakiego scenariusza testowego.

3.3. 3. Dla każdej podatności/błędu co najmniej informacja o
3.3.1. Należy odnieść wykryte podatności/błędy do rejestru znanych podatności (CVE.mitre.org oraz inne rejestry, w tym informacje producentów bibliotek i oprogramowania gotowego, stanowiącego część testowanego systemu) lub przekazanej listy pozycji do przetestowania wg ASVS level 1.
3.3.2. Matryca wynikowa ASVS w postaci tabeli excel (wzór w Załączniku nr 1) – z informacją przetestowano/nie przetestowano oraz komentarz z jakiego powodu, wynik testu – pozytywny/ negatywny.
3.3.3. Poziom ryzyka dla wykrytych podatności oznaczamy poziomem krytyczności, opisem oraz kolorem, w widoczny sposób. Nie stosujemy dużych ikonek.
3.3.4. termin wykrycia – data i czas,
3.3.5. nazwa serwera i jego IP,
3.3.6. odniesienie podatności do reguły OWASP ASVS, na podstawie których została wykryta podatność (jednej reguły lub kilku).

4. Załączniki

- Załącznik nr 1 – wzorcowa lista kontrolna ASVS: Wymagania ASVS.xlsx