

Zadanie: System Web Application Firewall

Założenia ogólne

Przedmiotem zamówienia jest dostarczenie i wdrożenie u Zamawiającego systemu bezpieczeństwa zapewniającego ochronę dla warstwy 7 (aplikacji) min. przed atakami ze strony botów i innych zautomatyzowanych narzędzi, atakami DoS.

Czas trwania umowy: wsparcie dla licencji wieczystej oraz moduły subskrypcyjne mają obowiązywać 12 miesięcy

W ramach zamówienia Wykonawca

- dostarczy funkcjonujący bezpieczeństwa zapewniającego ochronę dla warstwy 7 (aplikacji),
- wdroży i w pełni skonfiguruje dostarczony system dla wskazanych przez Zamawiającego 5 aplikacji.
- będzie świadczył usługi wsparcia dla działania systemu w trakcie trwania umowy (max. 40 godzin roboczych).

Własności ogólne Systemu:

1. Rozwiązanie musi realizować następujące funkcje:
 - a. rozkład ruchu pomiędzy serwerami aplikacji Web;
 - b. terminowanie sesji SSL;
 - c. optymalizacja i akceleracja aplikacji;
 - d. Wysoka dostępność i analityka;
 - e. Ochrona przed atakami na aplikacje internetowe i serwery WWW (Web Application Firewall) w tym ochrona przed atakami DDoS oraz ruchem zautomatyzowanym (BOT).
2. Przedmiotowy System musi być rozwiązaniem działającym w środowisku wirtualnym posiadanym przez Zamawiającego.
3. Przepustowość Systemu powinna wynosić minimum 1 Gbps z możliwością zwiększenia poprzez dokupienie licencji.
4. Licencja na przedmiotowe Rozwiązanie ma być licencją wieczystą z wykupywanym okresowo wsparciem Producenta rozwiązania, a ważność początkowego wsparcia Producenta powinna wynosić co najmniej 12 miesięcy.
5. W obrębie wsparcia Producenta zawarte musi być:
 - a. dostęp do aktualnych wersji oprogramowania oraz dokumentacji Producenta;
 - b. sposób obsługi zgłoszeń gwarancyjnych w trybie 7x24.
6. Musi istnieć możliwość odnowienia wsparcia Producenta po jego wygaśnięciu na okres co najmniej kolejnych 12 miesięcy.

7. Klucze prywatne zapisane na dysku przedmiotowego Rozwiązania wirtualnego muszą być zaszyfrowane. Nie dopuszcza się rozwiązań przechowujących klucze prywatne w formie jawnej.
8. Oferowany System będący rozwiązaniem wirtualnym powinien działać w następujących środowiskach wirtualnych oraz chmurowych:
 - a. VMware ESXi 5.5, 6.0-6.7 U1-U3, 7.0 U1-U3, 8.0 U3b, 8.0 U2, 8.0 U1a, 8.0, 7.0 U3c, 7.0 U3g
 - b. vCloud Director 5.5, 8.x, 9.x, v10.x;
 - c. Microsoft Hyper-V dla Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2 and Update 1, Windows Server 2012, Windows Server 2008 R2 SP1;
 - d. Linux KVM, Xen Project oraz OpenStack dla CentOS/RHEL od 6.3, Ubuntu od 14.04, Debian od 7.2;
 - e. Nutanix Acropolis Hypervisor (AHV) AOS 5.20.4.5 and AOS 6.5 LTS
 - f. Amazon Web Services;
 - g. Microsoft Azure;
 - h. Google Cloud Platform.
9. Wszystkie wymienione poniżej funkcje muszą być dostępne w obrębie jednego Rozwiązania wirtualnego.

Szczegółowe wymagania dla funkcji Web Application Firewall (WAF):

1. WAF musi posiadać możliwość działania w co najmniej dwóch modelach:
 - a. negatywnego modelu bezpieczeństwa (tylko to co szkodliwe jest blokowane)
 - b. pozytywnego modelu bezpieczeństwa (tylko to, co znane i prawidłowe jest dozwolone).
2. WAF musi co najmniej wspierać następujące tryby pracy:
 - a. tryb wykrywania, logowania i blokowania ataków;
 - b. tryb wykrywania i logowania ataków bez blokowania;
 - c. tryb uczenia się bez blokowania;
 - d. tryb uczenia się z blokowaniem i logowaniem.
3. WAF działając w oparciu o pozytywny model bezpieczeństwa, musi mieć możliwość utworzenia polityki ochrony na bazie automatycznie budowanego przez WAF profilu aplikacji Web. Pozytywny model bezpieczeństwa powinien kontrolować co najmniej:
 - a. wystąpienie URL-i, długość URL-i, zabezpieczenie przed clickjackiem dla danego URL-a.
 - b. typ servleta występujący pod danym url-em – format komunikacji (http form, JSON, XML, GWT)
 - c. przejścia pomiędzy URL-ami (servletami)
 - d. dopuszczalne metody http,
 - e. dopuszczalne cookie,
 - f. dopuszczalne parametry w polityce,
 - g. parametry dynamiczne,
 - h. typ/format parametrów (alfanumeryczny, integer, dynamiczny, statyczny, JSON, XML, e-mail, telefon, plik uploadowany)
 - i. oraz dopuszczalne parametry w danym serwlecie
 - j. długość zapytań



- k. nazwy hosta
 - l. wystąpienie i długość parametrów (per każdy parametr)
 - m. wystąpienie i długości nagłówków
 - n. wystąpienie i długości cookies
 - o. oczekiwanych typów znaków per każdy parametr
 - p. typów rozszerzeń plików; w tym długości URLa, requestu, query stringu, post data dla danego typu pliku
 - q. URL-i podatnych na CSRF
4. Profil aplikacji web musi być tworzony na podstawie analizy ruchu sieciowego.
 5. WAF musi umożliwiać definiowania dopuszczalnego przepływu sekwencji zapytań w obrębie aplikacji z uwzględnieniem jej logiki biznesowej.
 6. Oprócz pozytywnego modelu zabezpieczeń WAF musi posiadać również funkcje identyfikacji incydentów poprzez sygnatury (negatywny model zabezpieczeń).
 7. Tworzenie profilu bezpieczeństwa Web Application Firewall dla danej aplikacji musi odbywać się na podstawie analizy ruchu sieciowego.
 - a. W szczególności na podstawie publicznego ruchu produkcyjnego.
 - b. Algorytmy tworzenia profilu bezpieczeństwa WAF powinny odrzucać nadużycia w procesie nauki.
 8. Musi istnieć możliwość definicji zaufanych adresów źródłowych, z których algorytm tworzenia profilu bezpieczeństwa WAF będzie akceptować wszystkie zachowania jako prawidłowe, tak aby administrator mógł przyspieszyć proces tworzenia profilu bezpieczeństwa.
 9. Musi istnieć możliwość selektywnego włączania/wyłączania sygnatur per parametr.
 10. Musi istnieć możliwość ręcznego konfigurowania/modyfikacji reguł polityki bezpieczeństwa.
 11. Musi istnieć możliwość ochrony dynamicznych oraz ukrytych parametrów zapytań http.
 12. WAF musi posiadać funkcjonalność automatycznego wykrywania stron logowania użytkowników oraz automatycznie włączać dla tych stron ochronę przed atakami brute force.
 13. WAF musi posiadać mechanizmy ochrony przed atakami:
 - a. SQL Injection,
 - b. Cross-Site Scripting,
 - c. Cross-Site Request Forgery,
 - d. Session hijacking,
 - e. Command Injection,
 - f. Cookie/Session Poisoning,
 - g. Parameter/Form Tampering,
 - h. Forceful Browsing,
 - i. Brute Force Login,
 - j. Web Scraping
 - k. Cookie manipulation/poisoning
 - l. Dynamic Parameter tampering
 - m. Buffer Overflow
 - n. Stealth Commanding
 - o. Unused HTTP Methods
 - p. Malicious File Uploads
 - q. Hidden Field Manipulation
 14. Mechanizm zabezpieczenia przed manipulacją cookie serwera aplikacyjnego powinien być oparty o wstrzykiwanie cookie z podpisem oryginalnego cookie aplikacji.

15. Mechanizm zabezpieczenia przed Cross-Site Request Forgery powinien dodawać losowy token do odpowiedzi http zawierających odwołania do chronionego zasobu (servleta).
16. Wstrzykiwanie przez WAF dodatkowych informacji (cookie, tokeny, JavaScript), nie powinno powodować degradacji wydajności oferowanego Rozwiązania.
17. System musi zapewniać możliwość wyboru polityki bezpieczeństwa na podstawie:
 - a. Host
 - b. URN
 - c. Nagłówków
 - d. Cookie
18. Dla każdej chronionej aplikacji internetowej Rozwiązanie powinno umożliwiać wybór stosowanych technologii i systemu operacyjnego w celu poprawnego doboru wykorzystywanych sygnatur uwzględniając, ale nie ograniczając się do:
 - a. Bazy danych: ORACLE, MySQL, Microsoft SQL Server, PostgreSQL, Sybase, IBM DB2
 - b. System Operacyjny: Windows, Linux, UNIX
 - c. Język aplikacji, frameworki: ASP, ASP .NET, PHP, Java, BEA WebLogic, CGI, Elasticsearch, Front Page Server Extension, Java Servlets/JSP, Lotus Domino, Macromedia ColdFusion, JRun, Outlook Web Access, SSI, WebDAV, JQuery, SSI, WebDAV, JQuery
 - d. Serwer WWW: Apache, Apache Tomcat, Microsoft IIS, serwerów proxy.
19. WAF musi posiadać mechanizmy ochrony przed atakami DoS ukierunkowanymi na warstwę aplikacyjną (zalewanie aplikacji web dużą ilością zapytań http).
20. WAF musi blokować ataki typu Slow Loris.
21. WAF musi rozróżniać rzeczywistych użytkowników od automatów podczas ataku (D)DoS poprzez:
 - a. Wstrzykiwanie skryptu JavaScript i weryfikacji rezultatów jego wykonania
 - b. Mechanizmu browser fingerprinting, w celu wykrycia tzw. headless browser
 - c. Sygnatur botów
 - d. Wykorzystanie CAPTCHA (tylko w przypadku, gdy powyższe mechanizmy nie rozstrzygają czy podłączony jest rzeczywisty użytkownik).
22. System powinien umożliwiać proaktywne wykrywanie i blokowanie botów (j.w.), zanim wywołają atak DDoS, web scraping lub brute force.
23. System musi mieć możliwość nauczania się prawidłowego ruchu do aplikacji i na podstawie behawioralnej heurystyki chronić aplikację przed atakiem DDoS w warstwie 7, automatycznie budując regułę, która zablokuje atak oraz atakujące adresy IP. W systemie nie może być żadnego licencyjnego limitu dla tej funkcji.
24. System powinien kategoryzować boty i umożliwiać przepuszczanie ruchu od pożytecznych botów (np. search engine), blokując ruch od szkodliwych botów.
25. Moduł ochrony przed DDoS powinien wykrywać ataki per:
 - a. Source IP,
 - b. Obszar geolokacyjny,
 - c. URL,
 - d. Globalnie - website
26. Powinna istnieć możliwość przypisania różnych poziomów detekcji ataków (D)DoS dla danych URL-i portalu. Np. /infoportal/* powinien posiadać luźniejszą politykę detekcji i zapobiegania ataków DDoS niż /portal/*.

27. System powinien wykrywać i chronić przed atakami DDoS na tzw. ciężkie serwlety, czyli serwlety wywołujące złożone operacje obliczeniowe np. skomplikowane zapytania do baz danych.
 - a. Wykrycie ataku na ciężkie serwlety powinno opierać się przynajmniej o ilość zapytań (TPS) oraz czas odpowiedzi.
28. System powinien umożliwiać automatyczny zapis przykładowego ruchu do plików zgodnych z formatem TCP dump, w momencie wykrycia ataku (D)DoS:
 - a. System powinien umożliwiać definicję maksymalnego czasu próbki ruchu,
 - b. Maksymalnej pojemności próbki ruchu,
 - c. Interwału czasowego pomiędzy pobieraniem próbki ruchu.
29. Powinna istnieć możliwość doboru odpowiedzi w zależności do rodzaju naruszenia.
30. WAF musi posiadać możliwość uwzględniania w logach dotyczących incydentów informacji o uwierzytelnionym użytkowniku oraz blokowania dużej ilości incydentów wykonywanych w zdefiniowanym czasie przez tego użytkownika.
31. WAF powinien umożliwiać usuwanie nagłówków serwera aplikacyjnego zdradzających technologię oraz wersję oprogramowania; bez uszczerbku na wydajności WAF-a.
32. WAF powinien umożliwiać wstrzykiwanie nagłówków np. w celu ochrony przed Clickjack-iem.
33. WAF powinien umożliwiać podmianę kodów statusów zwracanych przez serwer aplikacyjny; bez uszczerbku na wydajności WAF-a.
34. W obrębie licencji WAF dostarczony musi być moduł ochrony protokołu HTTP, SMTP oraz FTP.
35. System musi być dostarczony z dedykowanym SDK WAF do aplikacji mobilnych, które musi oferować funkcję identyfikacji aplikacji mobilnej w systemie WAF odróżniając ją w ten sposób od podszywających się aplikacji/automatów.
36. Integracja aplikacji mobilne z SDK WAF musi odbywać się na zasadzie łączenia binarnego, bez konieczności modyfikacji kodu źródłowego aplikacji.
37. WAF musi posiadać wsparcie dla aplikacji AJAX oraz JSON.
38. WAF powinien wyświetlać stron blokowania (błędu) w technologiach AJAX i JSON.
39. WAF musi posiadać wsparcie dla Google Web Toolkit.
40. WAF musi posiadać możliwość ochrony komunikacji XML poprzez:
 - a. walidację Schema/WSDL,
 - b. wybór dozwolonych metod SOAP,
 - c. szyfrację /deszyfrację fragmentów wiadomości SOAP,
 - d. Wsparcie dla WS-Security (szyfracja, deszyfracja, verifyfikacja i podpisywanie),
 - e. Definiowanie możliwości użycia załączników wiadomości SOAP,
 - f. Włączanie/wyłączanie podążania za odnośnikami do schematów SOAP,
 - g. Walidację SOAPAction Header,
 - h. Włączanie/wyłączanie możliwości użycia DTD
 - i. Włączanie/wyłączanie możliwości użycia zewnętrznych referencji
 - j. Włączanie/wyłączanie możliwości użycia początkowych białych znaków
 - k. Włączanie/wyłączanie możliwości użycia numerycznych nazw
 - l. Włączanie/wyłączanie możliwości użycia Processing Instructions
 - m. Włączanie/wyłączanie możliwości użycia CDATA
 - n. Ograniczenie długości: dokumentu, elementu, nazwy, wartości atrybutu, Namespace
 - o. Ograniczenia ilości: zagnieżdżeń w dokumencie, dzieci per element, atrybutów per element, deklaracji Namespace-ów
 - p. Definicję dopuszczalnych znaków

- q. Definicję sygnatur.
41. WAF musi umożliwiać blokowanie zapytań z danego obszaru geograficznego. Aktualizacje bazy geolokacyjnej powinny być dostępne w ramach podstawowych opłat wsparcia.
 42. WAF musi umożliwiać automatyczne budowanie polityk w oparciu o skanowanie przez zewnętrznych dostawców, przynajmniej trzech, np. Cenzic, HP WebInspect, IBM AppScan, Qualys Guard, WhiteHat Sentinel.
 43. WAF musi posiadać mechanizmy normalizacji w celu obrony przed technikami ukrywania ataku. Mechanizmy normalizacji muszą wspierać/wykrywać:
 - a. Directory traversal
 - b. Kodowanie typu %u
 - c. Kodowanie typu IIS backslash
 - d. IIS Unicode codepoints
 - e. Bare byte decoding
 - f. Apache whitespace
 - g. Bad unescape
 - h. Wstrzykiwanie komentarzy (np. <!-- -->)
 44. Mechanizm normalizacji powinien umożliwiać definiowanie maksymalnego zagnieżdżonego kodowania.
 45. WAF musi umożliwiać integracje systemami antywirusowymi po protokole ICAP w celu wykrywania wirusów w przesyłanych plikach.
 46. WAF musi wykrywać i maskować numery kart kredytowych, wyciekających z chronionej aplikacji; oraz dowolnie inny ciąg znaków zdefiniowany poprzez PCRE regular expression.
 47. WAF musi chronić ruch przesyłany po IPv6 bez degradacji wydajności wynikającej z innych czynników niż różnice protokołów IPv4 i IPv6.
 48. System musi umożliwiać szyfrowanie wskazanych pól (np. pole do wprowadzania danych typu hasło) w czasie rzeczywistym, wprowadzanym w przeglądarce internetowej.
 49. Szyfrowanie musi być również dostępne, jeżeli formularz logowania wykorzystuje technologię AJAX.
 50. Szyfrowanie tych pól musi odbywać się z wykorzystaniem klucza publicznego osadzanego przez Rozwiązanie w odpowiedzi serwera aplikacyjnego. System nie może wymagać zmiany po stronie samej aplikacji ani wymagać instalacji dodatkowego oprogramowania na urządzeniu końcowym.
 51. System musi umożliwiać szyfrowanie w czasie rzeczywistym nazw wskazanych pól w kodzie HTML oraz dodawać dodatkowe pola typu input, by strona logowania www nie była statyczna (dodawanie dodatkowych pól typu input musi być niewidoczne dla użytkownika końcowego na stronie www).
 52. System powinien umożliwiać wykorzystanie funkcji sprawdzania reputacji adresów IP dostających się do chronionych aplikacji.
 53. Serwis reputacyjny adresów IP powinien być dostępny jako rozszerzenie systemu, bez konieczności wprowadzania zmian w architekturze wirtualnej oraz programowej proponowanego Rozwiązania.
 54. Licencja na serwis reputacyjny adresów IP musi zostać dostarczona wraz z przedmiotowym Systemem, a jego ważność powinna wynosić minimum 12 miesięcy.
 55. System powinien umożliwiać wykorzystanie usługi zawierającej kontekstowe informacje o charakterze i celu aktywnej kampanii zagrożeń.



Fundusze Europejskie
na Rozwój Cyfrowy



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



56. Sygnatury kontekstowe powinny być dostępne jako rozszerzenie systemu bez wprowadzania zmian w architekturze proponowanego Rozwiązania.
57. Licencja na serwis sygnatur kontekstowych musi zostać dostarczona wraz z przedmiotowym Systemem, a jego ważność powinna wynosić minimum 12 miesięcy.

Warunki udziału w postępowaniu

Wykonanie co najmniej dwóch zamówień o wartości minimum 80 000 zł brutto każde w zakresie dostawy i produkcyjnego wdrożenia systemu bezpieczeństwa zapewniającego ochronę dla warstwy 7 (aplikacji).

Wykonawca spełni warunek jeżeli wykaże w wykazie wykonanych, a w przypadku świadczeń okresowych lub ciągłych również wykonywanych, usług w okresie ostatnich 3 lat przed upływem terminu składania ofert, a jeżeli okres prowadzenia działalności jest krótszy – w tym okresie, że wykonał należycie co najmniej dwa zamówienia o wartości minimum 80 000 zł brutto każde w zakresie dostawy i produkcyjnego wdrożenia systemu bezpieczeństwa zapewniającego ochronę dla warstwy 7 (aplikacji).

Kary

Wykonawca za odstąpienie od umowy, niewykonanie lub nienależyte wykonanie zapisów umowy przewiduje zastosowanie kar powszechnie stosowanych dla zamówień tego typu.