Teoria

Atak typu Remote Code Execution (RCE) jest jednym z najbardziej niebezpiecznych rodzajów ataków, które pozwalają cyberprzestępcy na zdalne wykonywanie dowolnego kodu na zainfekowanym systemie. Tego rodzaju atak daje atakującemu pełną kontrolę nad systemem, co może prowadzić do poważnych naruszeń bezpieczeństwa, utraty danych czy przejęcia infrastruktury. Atak RCE polega na tym, że atakujący wykorzystuje lukę w zabezpieczeniach aplikacji, systemu operacyjnego lub aplikacji webowej, aby uruchomić złośliwy kod na zdalnym serwerze. Celem ataku jest zdobycie dostępu do systemu i uruchomienie kodu, który może wykonywać dowolne operacje, takie jak instalowanie malware, przechwytywanie danych lub eskalacja uprawnień.

Rodzaje ataków RCE:

- **Wykorzystanie niezweryfikowanych danych wejściowych:** Atakujący może wysłać złośliwe dane (np. w formularzu webowym), które zostaną następnie wykorzystane do wykonania złośliwego kodu na serwerze.
- **Wykorzystanie niewłaściwie skonfigurowanych usług:** Usługi, które pozwalają na zdalne wykonywanie kodu (np. serwery HTTP, serwery baz danych) mogą być źle skonfigurowane, umożliwiając nieautoryzowany dostęp do systemu.
- **Luki w oprogramowaniu:** Błędy w oprogramowaniu, które umożliwiają atakującym wysyłanie złośliwego kodu przez API, porty otwarte w systemie czy błędy w parsowaniu danych wejściowych.
- **Deserializacja obiektów:** Atakujący może wykorzystać mechanizm deserializacji obiektów do uruchomienia złośliwego kodu, gdy aplikacja źle obsługuje dane.

Skutki ataku RCE mogą być katastrofalne, w tym:

- **Zdalne przejęcie kontroli nad systemem:** Atakujący zyskuje dostęp do systemu i może wykonać dowolne operacje.
- **Kradzież danych:** Atakujący może uzyskać dostęp do wrażliwych danych, takich jak hasła, dane osobowe czy informacje o transakcjach.
- **Złośliwe oprogramowanie:** Możliwość zainstalowania złośliwego oprogramowania, takiego jak ransomware, keyloggery, boty do ataków DDoS.
- **Eskalacja uprawnień:** Atakujący może podnieść swoje uprawnienia do poziomu administratora lub root, co daje pełną kontrolę nad systemem.

Obrona przed atakami RCE

Obrona przed atakami RCE wymaga stosowania różnych strategii i najlepszych praktyk bezpieczeństwa w procesie tworzenia, wdrażania i utrzymywania oprogramowania. Poniżej przedstawiono kluczowe techniki obrony. Sanityzacja danych wejściowych to jeden z podstawowych sposobów zapobiegania atakom RCE. Ważne jest, aby dane dostarczane przez użytkowników były dokładnie sprawdzane przed ich wykorzystaniem w aplikacjach. Należy szczególną uwagę zwrócić na:

- Sprawdzanie, czy dane są odpowiedniego typu.
- Ograniczanie rozmiaru danych wejściowych.
- Usuwanie potencjalnie niebezpiecznych znaków, takich jak te używane w poleceniach systemowych.

Implementacja skutecznych mechanizmów kontroli dostępu pozwala ograniczyć możliwości atakującego. Do metod zapobiegania atakom RCE należą:

- Zastosowanie zasad minimalnych uprawnień (Principle of Least Privilege).
- Ograniczenie dostępu do funkcji zdalnego wykonania kodu do zaufanych użytkowników.
- Stosowanie firewalla i segmentacji sieciowej, aby zminimalizować dostęp do krytycznych zasobów.

Regularne aktualizowanie systemów operacyjnych, aplikacji i frameworków jest kluczowe w zapobieganiu atakom RCE. Większość ataków RCE wynika z wykorzystania znanych, ale niezałatanych luk w zabezpieczeniach. Regularne aktualizacje pomagają w zabezpieczeniu systemu przed nowymi zagrożeniami.

Zabezpieczenia warstwowe - Wielowarstwowa obrona jest kluczowa w zapobieganiu atakom. Obejmuje to:

- Użycie narzędzi do monitorowania aktywności w sieci i wykrywania nieautoryzowanych prób zdalnego wykonania kodu.
- Zastosowanie systemów wykrywania i zapobiegania włamaniom (IDS/IPS).
- Wykorzystanie sandboxów do izolowania potencjalnie złośliwych operacji.

Bezpieczne praktyki programowania - Programiści powinni stosować bezpieczne praktyki w procesie tworzenia oprogramowania, takie jak:

- Stosowanie odpowiednich metod kodowania, takich jak `prepare statement` w przypadku baz danych.
- Unikanie niebezpiecznych funkcji, które umożliwiają zdalne wykonanie kodu (np. `eval()` w JavaScript).
- Używanie bibliotek i frameworków z wbudowanymi mechanizmami ochrony przed atakami RCE.

Linki

- https://www.cloudflare.com/learning/security/what-is-remote-code-execution/
- https://www.invicti.com/learn/remote-code-execution-rce/
- https://www.splunk.com/en_us/blog/learn/rce-remote-code-execution.html
- https://www.vaadata.com/blog/rce-remote-code-execution-exploitations-and-security-tips/
- https://www.first.org/cvss/calculator/3.1
- https://blog.askomputer.pl/exiftool/
- https://magazynt3.pl/exif-i-iptc-metadane-opisujace-obrazy-cyfrowe/
- https://www.mankier.com/1/pngcrush
- https://sekurak.pl/jak-przez-upload-zwyklego-pliku-png-mozna-czytac-dowolne-pliki-z-serwera
- https://wojtek-m.blogspot.com/2010/07/magiczne-liczby.html

Pomocne narzędzia

Standardy EXIF oraz IPTC umożliwiają zapis metadanych w obrazach cyfrowych. Plik graficzny (np. JPEG) może zawierać - oprócz samej fotografii - informacje o autorze, słowa kluczowe, opis oraz parametry i ustawienia aparatu. Aparaty cyfrowe automatycznie zapisują w każdej wykonanej fotografii wiele informacji technicznych. Pliki JPEG, TIFF oraz RAW mogą zawierać, oprócz samego zdjęcia, także:

- dane aparatu: producenta i model,
- informacje o oprogramowaniu aparatu (np. wersję oprogramowania firmware),
- datę i godzinę wykonania,
- ustawienia aparatu: przysłonę, czas, tryb (manualny, automatyczny itd.), ogniskową itd.,
- opis zdjęcia,
- informacje o prawach autorskich,
- a nawet dane geolokacyjne pochodzące z odbiornika GPS.

Informacje EXIF zawarte w pliku JPG mogą być odczytane przy użyciu wielu programów, m.in. IrfanView, ExifRead, wtyczki FxIFFirefoksa oraz wbudowanych możliwości systemu. Dane EXIF mogą być odczytywane i modyfikowane z poziomu konsoli systemowej. Służy do tego np. program **exiv2**. Program ten działa na bazie biblioteki exiv2 dostępnej dla języka C++.

Magic numbers - `magicznie` wybrane liczby to stały element formatów plików czy protokołów - pełnią one najczęściej rolę identyfikatorów. Sam termin `magic number` ma swoje początki w systemie operacyjnym Unix w wersji 7 (rok 1979). Już w wersji szóstej Uniksa pierwsze 2 bajty programu wykonywalnego musiały mieć

wartość `0x0107`. W kolejnej wersji tego systemu wartość ta była podczas odczytywania pliku zapisywana w zmiennej o nazwie `ux_mag`. Zmienna ta dzięki dziwnej nazwie zyskała przydomek `magicznej liczby`. Dla innych typów plików zmienna ta przyjmowała inne wartości.

Gdy system operacyjny lub aplikacja chce określić typ pliku, może szukać na początku pliku specjalnego znacznika, który oznacza typ pliku. Na przykład plik PDF może zaczynać się od wartości szesnastkowej `0x255044462D312E35`, która równa się `% PDF-1.3` w formacie ASCII, lub plik ZIP zaczyna się od `0x504B`, co jest równe `PK`, co pochodzi od oryginalnego narzędzia `PKZip`. Patrząc na ten podpis, można łatwo zidentyfikować typ pliku, nawet bez innych metadanych.

exiftool - to potężne narzędzie linii poleceń, które umożliwia odczyt, zapis i edycję metadanych zawartych w plikach graficznych, dźwiękowych i wideo. Metadane to informacje dodatkowe zapisane w plikach, które zawierają różnorodne informacje na temat zawartości i historii pliku.

Metadane w plikach graficznych mogą zawierać informacje o aparacie, takie jak producent, model, ustawienia ekspozycji, ogniskowa, balans bieli itp. Można również znaleźć informacje dotyczące autorstwa, tytułów, dat utworzenia czy opisów. W przypadku plików dźwiękowych, metadane mogą obejmować informacje o albumie, artyście, gatunku, roku wydania i wielu innych szczegółach. W plikach wideo można znaleźć metadane dotyczące kodeków, rozdzielczości, długości, daty nagrania i innych danych technicznych.

exiftool obsługuje szeroki zakres formatów plików, obejmując popularne formaty obrazów, takie jak JPEG, TIFF, PNG, a także formaty RAW aparatów cyfrowych. Ponadto, obsługuje formaty dźwiękowe, takie jak MP3, WAV, FLAC, formaty wideo, takie jak AVI, MPEG, MOV, oraz wiele innych formatów.

Jedną z głównych zalet **exiftool** jest jego wszechstronność i możliwość manipulowania różnymi rodzajami metadanych. Narzędzie umożliwia odczytanie i wyświetlenie metadanych. Umożliwia także edycję istniejących tagów, dodawanie nowych tagów, usuwanie tagów i kopiowanie metadanych między plikami. Możliwość dokładnej manipulacji metadanymi pozwala użytkownikom na dostosowanie informacji w plikach zgodnie z ich potrzebami. **exiftool** jest narzędziem wiersza poleceń, co oznacza, że korzysta się z niego poprzez wprowadzanie poleceń tekstowych w terminalu lub wierszu polecenia. Może być używany na różnych systemach operacyjnych, takich jak Windows, macOS i Linux, co czyni go elastycznym i przenośnym narzędziem.

exiv2 - narzędzie do odczytywania i modyfikowania dodatkowych danych zapisanych w pliku.

xxd - komenda xxd w systemie Linux pozwala utworzyć zrzut heksadecymalny, a nawet wykonać odwrotną czynność.

hexeditor - The HexEdit Hex Editor is another Hex Editor used for editing binary files. Unlike the Xxd Hex Editor, the ASCII (numerical coding) form of the file is also shown by HexEdit. For modern operating systems, including Linux and Windows, this editor is mostly used.

Zadania

10.1 *HTTP server running on IP address `127.0.0.1` and port `10001` is vulnerable to Arbitrary File Upload, which can result in Remote Code Execution (RCE). HTTP server is using HTTP/1.1 protocol. It allows for uploading files, which are uploaded to `http://127.0.0.1:10001/uploads`.*

(a) Zapoznaj się z opisem podatności Arbitrary File Upload.

(b) Korzystając z przygotowanego obrazu Dockerowego (mazurkatarzyna/cik-book-p1-ch10-ex1:latest), uruchom podatny serwer HTTP. Serwer pozwala na upload plików, które po wysłaniu na serwer znajdują się w katalogu `http://127.0.0.1:10001/uploads`.

(c) Wysyłając na serwer plik z rozszerzeniem `*.php` doprowadź do wyświetlenia zawartości pliku `/etc/passwd` znajdującego się na serwerze.

10.2 *HTTP server running on IP address `127.0.0.1` and port `10002` is vulnerable to Arbitrary File Upload, which can result in Remote Code Execution (RCE). HTTP server is using HTTP/1.1 protocol. It allows for uploading files, which are uploaded to `http://127.0.0.1:10002/uploads`.*

(a) Zapoznaj się z opisem podatności Arbitrary File Upload.

(b) Korzystając z przygotowanego obrazu Dockerowego (mazurkatarzyna/cik-book-p1-ch10-ex2:latest), uruchom podatny serwer HTTP. Serwer pozwala na upload plików, które po wysłaniu na serwer znajdują się w katalogu `http://127.0.0.1:10002/uploads`.

(c) Wysyłając na serwer plik z rozszerzeniem `*.png` doprowadź do wyświetlenia zawartości pliku `/etc/passwd` znajdującego się na serwerze oraz wyświetlenia informacji o konfiguracji PHP na serwerze (możesz użyć funkcji phpinfo();).

10.3 *HTTP server running on IP address `127.0.0.1` and port `10003` is vulnerable to Arbitrary File Upload, which can result in Remote Code Execution (RCE). HTTP server is using HTTP/1.1 protocol. It allows for uploading files, which are uploaded to `http://127.0.0.1:10003/uploads`.*

(a) Zapoznaj się z opisem podatności Arbitrary File Upload.

(b) Korzystając z przygotowanego obrazu Dockerowego (mazurkatarzyna/cik-book-p1-ch10-ex3:latest), uruchom podatny serwer HTTP. Serwer pozwala na upload plików, które po wysłaniu na serwer znajdują się w katalogu `http://127.0.0.1:10003/uploads`.

(c) Serwer zezwala jedynie na upload plików z rozszerzeniem `jpeg, png, jpg, pdf`. Omiń zabezpieczenia serwera i wyślij na serwer plik z rozszerzeniem `*.html`.

10.4 *`Cockpit` is a modern content platform - simple but yet powerful and flexible. Cockpit provides a straightforward way to manage content for various applications, especially when you need a flexible structure and a simple API to fetch content. Whether you're building a website, a mobile application, or a SPA, Cockpit can serve as a lightweight but flexible backend to power your content. In version `0.12.2` `Cockpit` is vulnerable to Remote File Inclusion (CVE-2023-4195) which results in Remote Code Execution (RCE).*

(a) Zapoznaj się z opisem podatności **CVE-2023-4195**.

(b) Korzystając z przygotowanego pliku Dockerowego (`docker-compose-cve-2023-4195.yml`), uruchom aplikację `Cockpit` w podatnej wersji `0.12.2`. Wraz z aplikacją `Cockpit` zostanie również uruchomiony serwer SSH.

(c) Udowodnij, że podatność w aplikacji `Cockpit` istnieje.

(d) Przeprowadź atak RCE na aplikacji `Cockpit`, dzięki któremu uzyskasz dostęp do serwera SSH.

(e) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

10.5 *Apache is the most widely used webserver software and runs on most websites in the world. Developed and maintained by Apache Software Foundation, Apache is open source software and available for free. The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. Apache HTTP Server in version 2.4.50 is vulnerable to Remote Code Execution.*

(a) Zapoznaj się z opisem podatności **CVE-2021-42013**

(b) Korzystając z przygotowanego obrazu Dockerowego (mazurkatarzyna/cve-2021-42013:latest), uruchom serwer `Apache`.

(c) Przeprowadź atak Remote Code Execution:

- Wyświetl zawartość katalogu `/etc` na serwerze
- Sprawdź nazwę oraz wersję zainstalowanego systemu operacyjnego na serwerze

Użyj narzędzi `telnet`, `cURL` oraz skryptu w języku Python.

(d) Po udanym przeprowadzeniu ataku, oszacuj poziom krytyczności podatności **CVE-2021-42013** używając kalkulatora CVSS.

10.6 *Jenkins is an open source automation server which enables developers around the world to reliably build, test, and deploy their software. In version 2.289.1 Jenkins is vulnerable to remote code execution (CVE-2019-1003000). A flaw was found in Jenkins Pipeline. The Script Security sandbox protection could be circumvented during the script compilation phase by applying AST, transforming annotations such as @Grab to source code elements. Both the pipeline validation REST APIs and actual script/pipeline execution are affected. This allowed users with Overall/Read permission, or able to control Jenkinsfile or sandboxed Pipeline shared library contents in SCM, to bypass the sandbox protection and execute arbitrary code on the Jenkins master. All known unsafe AST transformations in Groovy are now prohibited in sandboxed scripts. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.*

(a) Zapoznaj się z opisem podatności CVE-2019-1003000

(b) Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/cve-2019-1003000:latest`), uruchom kontener z serwerem, który wykorzystuje podatną wersję Jenkinsa.

(c) Wykorzystując podatność, przeprowadź atak RCE, aby odczytać zawartość pliku `/etc/passwd` znajdującego się na serwerze.

(d) Po udanym przeprowadzeniu ataku, oszacuj poziom krytyczności podatności używając kalkulatora CVSS.

10.7 *Wordpress is a web content management system. It was originally created as a tool to publish blogs. WordPress is an open-source content management system (CMS). The Advanced Uploader WordPress plugin through 4.2 allows any authenticated users like subscriber to upload arbitrary files, such as PHP, which could lead to RCE.*

(a) Zapoznaj się z opisem podatności **CVE-2022-1103**.

(b) Korzystając z przygotowanego pliku Dockerowego `docker-compose-cve-2022-1103.yml`, uruchom aplikację `Wordpress`. Pobierz plik `advanced-uploader.zip` ze strony przedmiotu, zainstaluj wtyczkę w Wordpressie.

(c) Wiedząc o istniejącej podatności wtyczki, wyślij na serwer plik, dzięki któremu odczytasz zawartość pliku `/etc/passwd` znajdującego się na serwerze.

(d) Przeprowadź atak RCE, dzięki któremu będziesz w stanie wykonać dowolną komendę na zdalnym serwerze ze swojego komputera.