آسیب‌پذیری Unserialization در وب‌سرور WebLogic

سازمان فناوری اطلاعات ایران

مـــرکـــز مـــاهـــر

دی ۹۶

۱- شرح آسیب پذیری:

بیشتر زبان‌های برنامه‌نویسی روش‌های Built-in را برای کاربران فراهم می‌کنند تا بتوانند داده‌های برنامه‌های کاربردی را بر روی دیسک ذخیره کننده و یا از طریق شبکه انتقال دهند. فرایند تبدیل داده‌های برنامه‌ی کاربردی به فرمت‌های مناسب جهت انتقال (معمولاً باینری) Serialization نامیده می‌شود و فرایند بازخواندن داده‌ها بعد از انجام Serialization، Deserialization/Unserialization نامیده می‌شود.

آسیب‌پذیری‌های مربوط به Unserialization، کلاسی از آسیب‌پذیری‌های پر خطر هستند که زمانی ایجاد می‌شوند که توسعه‌دهندگان کدهایی را ایجاد کنند که داده‌هایSerialized را از کاربران دریافت کنند و تلاش می‌کنند آن را برای استفاده در برنامه بدون پاک‌سازی Unserialize کنند. این آسیب‌پذیری بسته به زبان برنامه‌نویسی می‌توانند منجر به عواقب متفاوتی گردد، که یکی از مهم‌ترین عواقب مورد علاقه‌ی هکرها، اجرای دستور سیستم-عاملی از راه دور (RCE[۱]) است.

توضیحات بیشتر در مورد این نوع آسیب‌پذیری در مسیر زیر قابل مشاهده است:

https://www.owasp.org/index.php/Deserialization_of_untrusted_data

جاوا یکی از زبان‌های برنامه‌نویسی‌ای است که از Serialize کردن، جهت نوشتن و یا انتقال داده‌ها استفاده می‌کند و یا به تعبیر دیگر در جاوا هر جایی می‌توان از Object Serialization استفاده نمود و توابع راه گریزی از پذیرفتن داده‌های Serialized شده‌ی ناامنِ کاربر، ندارند. در سال‌های اخیر تعدادی آسیب‌پذیری بر روی چارچوب[۲]های مبتنی بر جاوا کشف شده است. یکی از آسیب‌پذیری‌های اخیر، مربوط به یکی از کتابخانه‌های بسیار مورد استفاده در چارچوب‌ها و برنامه‌های کاربردی مبتنی بر جاوا با کد CVE-۲۰۱۵-۴۸۵۲ است. در ۲۸ ژانویه ۲۰۱۵، آقایان Frohoff و Lawrence، دنیا را از وجود یک آسیب‌پذیری Unserialization بر روی یکی از کتابخانه‌های جاوا به نام "Commons Collections" که منجر به RCE می‌گردد، مطلع کردند که نسخه‌ی ۳,۲,۱ و ٤,۰ این کتابخانه دارای آسیب‌پذیری است. این کتابخانه یکی از کتابخانه‌های بسیار معروف در زبان جاوا

[۱] Remote Code Execution
[۲] Framework

است که تعداد زیادی ساختار داده‌ای تعریف کرده است که توسعه‌ی برنامه‌های مبتنی بر جاوا را تسریع می‌بخشد و تبدیل به یک استاندارد برای مدیریت Collection ها در جاوا شده است. هر برنامه‌های جاوا و یا زیر ساخت جاوا که از این کتابخانه استفاده و یا این کتابخانه را در Classpath اضافه کرده باشد، در صورتی که داده‌های دریافتی از کاربر را Unserialize کند، دارای این آسیب‌پذیری، با CVSS[۳] ۱۰,۰ است. این امر به این معنی است که هر کسی روی شبکه و اینترنت می‌تواند تعداد زیادی از سرورهای برنامه‌ی کاربردی آسیب‌پذیر، از جمله Appliance ها را تحت نفوذ خود درآورد. کلاس InvokerTransformer، کلاسی است که منجر به این آسیب‌پذیری شده است. این آسیب‌پذیری چندین زیرساخت بزرگ شامل WebSphere، JBoss، Jenkins، WebLogic، OpenScape و OpenNMS که از این کتابخانه استفاده کرده‌اند را تحت تاثیر قرار می‌دهد. سوءاستفاده از این آسیب‌پذیری زمانی میسر می‌گردد که یک کلاس از برنامه از Deserialization ناامن بر روی برخی از Stream های ورودی استفاده کند. بنابراین یک هکر می‌تواند Object دارای کد بدخواه[۴] را به Stream ارسال کند و بعد از انجام Deserialization بر روی مقادیر دریافتی منجر به اجرای کدهای بدخواه گردد.

۲–روش امن‌سازی

جهت برطرف کردن این آسیب‌پذیری دو راه حل اصلی و پنج راه حل فرعی (در صورت در دسترس نبودن و یا امکان پذیر نبودن راه حل اصلی) وجود دارد:

راه حل اصلی

۱. به‌روز رسانی WebLogic به آخرین نسخه موجود (و یا حداقل نسخه‌های فاقد آسیب‌پذیری)
۲. نصب patch ارائه شده توسط شرکت اوراکل:

https://support.oracle.com/rs?type=doc&id=۲۰۷۵۹۲۷,۱

اطلاعات بیشتر در لینک زیر قابل دسترس خواهد بود:

https://www.oracle.com/technetwork/topics/security/alert-cve-۲۰۱۵-٤٨٥٢-٢٧٦٣٣٣٣.html

[۳] Common Vulnerability Scoring System https://www.first.org/cvss
[۴] Malicious

راه حل فرعی

در صورت در دسترس نبودن patch از راهکارهای زیر استفاده نمایید:

۱. عدم پذیرش ارتباطات T۳ با استفاده از IPS و یا iptables و یا استفاده از ماژول تعبیه شده درون WebLogic به نام Network Connection Filter که مانند یک firewall عمل می‌کند. اطلاعات بیشتر در مورد این ماژول و نحوه‌ی فیلتر کردن این پروتکل، در لینک زیر قرار دارد:

https://docs.oracle.com/cd/E۲۳۹٤۳_۰۱/web.۱۱۱۱/e۱۳۷۱۱/con_filtr.htm#SCPRG۳۷۸

۲. حذف کلاس InvokerTransformer از درون کتابخانه‌ی آسیب‌پذیر commons-collections باشد. این کلاس در زمان حمله و ساختن exploit مورد استفاده قرار می‌گیرد. با استفاده از دستور زیر در محیط لینوکس می‌توان هر کلاس و یا jar که درون آن این کلاس استفاده شده است را یافت:

grep -Rl InvokerTransformer /

سپس باید تمامی آن‌ها را حذف کرد.
همچنین می‌توان کتابخانه‌ی آسیب‌پذیر commons-collections را extract کرده، در مسیر org/apache/commons/collections/functors/InvokerTransformer.class کلاس آسیب‌پذیر InvokerTransformer را یافته، آن را حذف و دوباره jar فایل را ایجاد کرد.

۳. در سومین روش باید ترافیک کنسول مدیریتی Weblogic فقط بر روی شبکه داخلی Allow باشد. در روشی دیگر می توان از وب سرور Apache Tomacat به منظور Reverse Proxy در جلوی وب سرور Weblogic قرارداد.

٤. چهارمین روش بر تغییر معماری Deployment وب سرور Weblogic استوار است. در این معماری از Apache Tomcat به منظور وب سرور در DMZ و از Weblogic به عنوان Application سرور در شبکه داخلی قرار داده می شود. سپس توسط firewall دسترسی به کنسول مدیریتی Weblogic محدود می شود.

۵. پنجمین روش بر مبنای غیرفعال کردن پروتکل T۳در وب سرورWeblogic می باشد. ابتدا با استفاده از لینک ذیل یک Connection Filtering راه اندازی کرده :

https://docs.oracle.com/cd/E۲۴۳۲۹_۰۱/web.۱۲۱۱/e۲۴۴۸۵/con_filtr.htm#SCPRG۳۷۷

سپس پورت پیش فرض را تغییر داده و یک کانال Http-only بر روی پورت پیش فرض با استفاده از لینک ذیل ایجاد می شود.

https://docs.oracle.com/cd/E۲۴۳۲۹_۰۱/web.۱۲۱۱/e۲۴۴۳۲/network.htm#CNFGD۱۵۹