به نام خدا

نرگس منتظری - 810103284

درس امنیت و حریم خصوصی - تکلیف دوم

مقدمه

این پروژه با هدف طراحی یک افزونه مرورگر برای بررسی گواهینامه‌های SSL/TLS و استخراج اطلاعات مهم از آن‌ها ایجاد شده است. افزونه قادر است اطلاعات گواهینامه سایت‌ها را استخراج کرده و به صورت گرافیکی در یک رابط کاربری کاربرپسند نمایش دهد.

ساختار این افزونه به صورت زیر می‌باشد که عملکرد فایل‌های مهم در ادامه توضیح داده شده است.

```
cert-checker/
├── manifest.json
├── background.js
├── popup/
│   ├── popup.html
│   ├── popup.js
│   └── popup.css
└── icons/
    └── icon.png
```

1. **manifest.json:** مدیریت تنظیمات و مجوزهای افزونه.
2. **background.js:** دریافت اطلاعات گواهینامه و پردازش داده‌ها.
3. **popup.html:** رابط کاربری برای نمایش اطلاعات به کاربر.
4. **popup.js:** کنترل تعاملات کاربر و نمایش داده‌ها در رابط.
5. **popup.css:** استایل‌دهی و طراحی رابط کاربری.

مراحل استخراج و پردازش اطلاعات

از API مرورگر به نام webRequest.getSecurityInfo برای دریافت اطلاعات گواهینامه استفاده شده است. این API امکان استخراج زنجیره گواهینامه‌ها (certificateChain) و اطلاعات خام (rawDER) را فراهم می‌کند.

```
const certDetails = await browser.webRequest.getSecurityInfo(
  details.requestId,
  {
    certificateChain: true,
    rawDER: false,
  }
);
```

داده‌های دریافت‌شده شامل اطلاعات مربوط به صادرکننده، موضوع، تاریخ شروع و پایان اعتبار، نوع کلید، الگوریتم امضای دیجیتال، و شماره سریال است. داده‌ها به صورت یک آبجکت ذخیره می‌شوند:

```
certDataCache = {
  issuerName: cert.issuer || "ناموجود",
  subjectName: cert.subject || "ناموجود",
  validFrom: cert.validity?.start || "ناموجود",
  validTo: cert.validity?.end || "ناموجود",
  keyAlgorithm: cert.subjectPublicKeyInfo?.algorithm || "ناموجود",
  sigAlgorithm: cert.fingerprint.sha256 || "ناموجود",
  serialNumber: cert.serialNumber || "ناموجود",
};
```

داده‌های پردازش‌شده از طریق پیام‌رسانی بین background.js و popup.js به رابط منتقل می‌شوند و سپس در یک جدول مرتب نمایش داده می‌شوند.

سپس برای استفاده از این پلاگین manifest.json را در قسمت about:debugging#/runtime/this-firefox در فایرفاکس وارد می‌کنیم. تا بتوانیم از این پلاگین استفاده کنیم. صفحه اول این پلاگین به صورت زیر است که می‌توان urlهای مختلف را در آن وارد کرد:

که مثال‌هایی از استفاده از این پلاگین به صورت زیر می‌باشد.

مثال ۱: (سایت google.com)

این گواهینامه متعلق به یکی از زیردامنه‌های گوگل به نام misc.google.com است که توسط Google Trust Services صادر شده است. تاریخ اعتبار این گواهینامه از 10 مارس 2025 آغاز و در 2 ژوئن 2025 به پایان می‌رسد. الگوریتم کلید به دلیل محدودیت API مشخص نشده است. امضای دیجیتال به درستی استخراج شده و با الگوریتمی مانند SHA-256 هش شده است.

مثال ۲: (سایت torob.com)

صادر کننده	CN=Certum Domain Validation CA SHA2,OU=Certum Certification Authority,O=Unizeto Technologies S.A.,C=PL
موضوع	CN=*.torob.com
تاریخ شروع اعتبار	AM 10:08:25 ,3/24/2025
تاریخ پایان اعتبار	AM 10:08:24 ,3/24/2026
الگوریتم کلید	ناموجود
امضای دیجیتال	CF:24:68:D3:68:7A:C8:F8:6A:8A:51:10:E9:B5:C0:3E:88:65:73:D8:C4:33:86:FA:F9:C3:49:DF:A7:76:3D:5F
شماره سریال	DE:1D:1E:AE:52:5C:4A:43:90:8:48F:6A:B1:D6:4B:E2

این گواهینامه Wildcard SSL توسط Unizeto Technologies S.A صادر شده است که یک مرجع صدور گواهینامه معتبر اروپایی است. تاریخ شروع اعتبار از 24 مارس 2025 است و این گواهینامه تا 24 مارس 2026 معتبر می‌باشد. گواهینامه از نوع Wildcard است که برای تمامی زیردامنه‌های ترب معتبر است. امضای دیجیتال معتبر است و تضمین می‌کند که گواهینامه تقلبی یا جعلی نیست. الگوریتم کلید مشخص نشده است که ممکن است به دلیل محدودیت API باشد.

مثال ۳: (دامنه نامعتبر)

مثال ۴: (عدم وجود گواهینامه)

پاسخ به سوالات انتهایی

۱. چرا کاربران عادی برای درک گواهینامه‌های X.509 با مشکل مواجه می‌شوند؟

الف) پیچیدگی ساختار گواهینامه

گواهینامه‌های X.509 معمولاً دارای ساختاری پیچیده و فنی هستند که برای کاربران عادی درک آن‌ها دشوار است. این گواهینامه‌ها شامل اطلاعات متعددی مانند صادر کننده (Issuer)، موضوع (Subject)، کلید عمومی (Public Key)، امضای دیجیتال (Digital Signature)، الگوریتم رمزنگاری و تاریخ اعتبار هستند.

هر یک از این مولفه‌ها با استفاده از نام‌های فنی (مانند CN، OU و O) نمایش داده می‌شوند که به طور مستقیم برای کاربران عادی قابل فهم نیستند. همچنین، اصطلاحات رمزنگاری مانند SHA-256 یا RSA-2048 بدون داشتن دانش تخصصی از علم رمزنگاری بسیار گمراه‌کننده به نظر می‌رسند.

ب) ماهیت رمزنگاری و امضاهای دیجیتال

کاربران عادی معمولاً با مفهوم امضاهای دیجیتال و نحوه تأیید هویت دیجیتال آشنا نیستند. برای آن‌ها این سوال مطرح است که چرا به یک گواهینامه باید اعتماد کنند و چگونه یک امضای دیجیتال می‌تواند امنیت ارتباط را تضمین کند. در واقع، فرایند اعتبارسنجی گواهینامه توسط مرورگرها و سیستم‌ها در پس‌زمینه انجام می‌شود و کاربران فقط با پیغام‌های خطا یا اخطار مواجه می‌شوند، بدون اینکه به طور کامل بدانند چرا چنین اتفاقی افتاده است.

پ) اعتماد به مراجع صدور گواهینامه (CA)

یکی دیگر از چالش‌های کاربران عادی این است که مرجع صدور گواهینامه (CA) چگونه قابل اعتماد است. اغلب کاربران درک نمی‌کنند که چرا یک مرجع مانند Let’s Encrypt یا DigiCert معتبر است و چه تفاوتی با یک گواهینامه خودامضا (Self-Signed) دارد. این عدم آگاهی می‌تواند به استفاده از سایت‌های ناامن منجر شود یا باعث بی‌اعتمادی بی‌مورد به سایت‌های امن گردد.

ت) پیغام‌های خطا و اخطارها

مرورگرها در صورت بروز مشکل در گواهینامه‌ها اخطارهای مبهمی مانند "اتصال شما خصوصی نیست" یا "گواهینامه نامعتبر است" نشان می‌دهند. این پیام‌ها معمولاً شامل جزئیات فنی هستند که کاربران عادی قادر به تفسیر آن‌ها نیستند. همین موضوع باعث می‌شود کاربران یا این هشدارها را نادیده بگیرند یا با ترس از ادامه کار صرف نظر کنند.

۲. مزایا و محدودیت‌های استفاده از فریمورک WebExtensions در مقایسه با سایر فریمورک‌های مشابه

- مزایای WebExtensions

یکی از بزرگ‌ترین مزایای WebExtensions این است که یک افزونه نوشته شده با این فریمورک می‌تواند به صورت یکپارچه روی چندین مرورگر از جمله Chrome، Firefox، Edge و Opera کار کند. این قابلیت باعث می‌شود که توسعه‌دهندگان تنها یک بار کد بنویسند و آن را برای چندین پلتفرم منتشر کنند، که به شدت زمان و هزینه توسعه را کاهش می‌دهد.

WebExtensions دارای APIهای استاندارد و مستندات کامل است که موجب می‌شود توسعه‌دهندگان به راحتی قابلیت‌هایی مانند مدیریت تب‌ها، درخواست‌های شبکه و دسترسی به حافظه محلی را پیاده‌سازی کنند. این یکپارچگی استاندارد، نگهداری و به‌روزرسانی افزونه‌ها را تسهیل می‌کند.

یکی دیگر از مزایای مهم WebExtensions، مدل امنیتی بهبود یافته است که از دسترسی مستقیم به منابع حساس جلوگیری می‌کند. مجوزها به صورت دقیق و محدود تعریف شده‌اند، که امکان سوءاستفاده از داده‌های کاربر را کاهش می‌دهد.

- محدودیت‌های WebExtensions

WebExtensions به منظور بهبود امنیت، محدودیت‌هایی در سطح دسترسی به منابع سیستمی و فایل‌های محلی دارد. این موضوع باعث می‌شود که افزونه‌ها برای انجام برخی وظایف پیچیده، نیاز به اجرای برنامه‌های بومی (Native) یا استفاده از Native Messaging داشته باشند.

برخلاف فریمورک‌های قبلی مانند XUL در Firefox، WebExtensions توانایی ایجاد تغییرات عمیق در رابط کاربری مرورگر یا مدیریت مستقیم ماژول‌های مرورگر را ندارد. این موضوع ممکن است توسعه‌دهندگان را که نیاز به شخصی‌سازی عمیق دارند، با محدودیت مواجه کند.
اگرچه WebExtensions به صورت چندمرورگری طراحی شده است، اما در عمل ممکن است برخی APIها در همه مرورگرها یکسان عمل نکنند. به عنوان مثال، API مدیریت تب‌ها در Firefox متفاوت از Chrome پیاده‌سازی شده است، که نیاز به کدنویسی شرطی برای هر مرورگر دارد.

۳. چه پارامترهایی در یک گواهینامه بیشترین اهمیت را در تضمین امنیت یک وب‌سایت دارند؟

الف) صادر کننده (Issuer): صادر کننده (CA) یکی از مهم‌ترین پارامترها در گواهینامه است. اعتبار یک وب‌سایت به شدت به اعتبار مرجع صدور گواهینامه (CA) وابسته است. مراجع معتبر مانند DigiCert، Let's Encrypt و GlobalSign تضمین می‌کنند که فرایند صدور گواهینامه با استانداردهای امنیتی دقیق انجام شده است. اگر گواهینامه توسط یک CA شناخته شده صادر شده باشد، می‌توان به اعتبار آن اطمینان داشت. اما اگر توسط یک CA نامعتبر یا خودامضا (Self-Signed) باشد، ممکن است خطرناک باشد.
ب) کلید عمومی (Public Key): گواهینامه‌ها معمولاً حاوی یک کلید عمومی RSA یا ECC هستند. طول و نوع کلید نشان‌دهنده سطح امنیت رمزنگاری است. به عنوان مثال، کلیدهای RSA با طول 2048 بیت یا ECC با طول 256 بیت به عنوان استاندارد امن شناخته می‌شوند.هرچه طول کلید بیشتر باشد، امنیت آن در برابر حملات جستجوی فراگیر (Brute Force) بالاتر است.
پ) الگوریتم امضا (Signature Algorithm): الگوریتم امضای دیجیتال، مانند SHA-256 با RSA، نشان‌دهنده میزان امنیت در تولید و اعتبارسنجی گواهینامه است. الگوریتم‌های قدیمی‌تر مانند SHA-1 به دلیل آسیب‌پذیری‌های کشف شده دیگر ایمن نیستند. استفاده از الگوریتم‌های مدرن مانند SHA-256 نشان‌دهنده امنیت بهتر گواهینامه است.
ت) تاریخ اعتبار (Validity Period): این پارامتر مشخص می‌کند که گواهینامه از چه تاریخی تا چه تاریخی معتبر است. گواهینامه‌های کوتاه‌مدت، مانند 90 روزه، به دلیل تجدید مداوم امن‌تر هستند، زیرا در صورت نشت کلید، مدت آسیب‌پذیری کوتاه‌تر است.اگر تاریخ اعتبار منقضی شده باشد، سایت دیگر ایمن نیست و ممکن است مورد حمله مرد میانی (MITM) قرار گیرد.
ث) نام موضوع (Subject): این پارامتر شامل اطلاعاتی مانند نام دامنه (CN) و سازمان مالک وب‌سایت است. عدم مطابقت بین نام دامنه و موضوع گواهینامه می‌تواند نشان‌دهنده یک حمله فیشینگ باشد. باید نام دامنه در گواهینامه با آدرس وب‌سایت مطابقت داشته باشد.

ج) زنجیره اعتبار (Certificate Chain): زنجیره اعتبار مشخص می‌کند که چگونه یک گواهینامه به مرجع ریشه (Root CA) متصل است. زنجیره باید به طور کامل و صحیح پیکربندی شده باشد تا مرورگر بتواند اعتبار آن را تأیید کند. در صورتی که زنجیره اعتبار ناقص باشد، مرورگر نمی‌تواند گواهینامه را تأیید کند و هشدار امنیتی نمایش می‌دهد.

۴. آیا استفاده از API های مرورگر جهت استخراج گواهینامه‌ها می‌تواند منجر به بروز مشکلات امنیتی شود؟

استفاده از APIهای مرورگر برای استخراج اطلاعات گواهینامه‌ها می‌تواند امنیت کاربران را به خطر بیندازد، به ویژه اگر کنترل‌های لازم برای جلوگیری از دسترسی غیرمجاز و افشای اطلاعات حساس اعمال نشود. به همین دلیل، توسعه‌دهندگان باید هنگام استفاده از این APIها به اصول امنیتی توجه کنند و از روش‌های استاندارد برای محافظت از داده‌ها بهره بگیرند. بعضی از این مشکلات در ادامه به طور کامل توضیح داده شده است:

الف) افشای اطلاعات حساس

APIهای مرورگر مانند Web Crypto API یا SSL Certificate API در برخی مرورگرها می‌توانند به گواهینامه‌ها دسترسی داشته باشند. اگر این APIها به درستی کنترل نشوند، می‌توانند اطلاعاتی مانند نام مرجع صادر کننده (Issuer)، تاریخ انقضا و حتی کلید عمومی را فاش کنند. این اطلاعات می‌تواند توسط مهاجمان برای مهندسی اجتماعی یا حملات فیشینگ مورد استفاده قرار گیرد.

ب) امکان حملات مرد میانی (MITM)

اگر APIهای مرورگر به طور مستقیم به گواهینامه‌های SSL دسترسی داشته باشند، ممکن است مهاجمان از طریق حملات مرد میانی داده‌ها را تغییر داده یا اطلاعات جعلی وارد کنند. به ویژه در شبکه‌های ناامن مانند Wi-Fi عمومی، این نوع حملات می‌توانند به راحتی انجام شوند.

پ) ریسک‌های مربوط به افزونه‌های غیرمجاز

برخی افزونه‌های مرورگر ممکن است از APIهای مرتبط با گواهینامه‌ها برای جمع‌آوری اطلاعات حساس استفاده کنند. افزونه‌های مخرب می‌توانند با استفاده از این APIها، داده‌های مربوط به گواهینامه‌های وب‌سایت‌های مورد بازدید را استخراج کرده و به سرورهای مخرب ارسال کنند.

ت)مشکلات امنیتی ناشی از پیاده‌سازی ناقص

برخی توسعه‌دهندگان ممکن است از این APIها به صورت ناایمن استفاده کنند، که می‌تواند به افشای غیرمجاز اطلاعات گواهینامه منجر شود. علاوه بر این، برخی از APIها به صورت پیش‌فرض اطلاعات دقیق گواهینامه را در اختیار همه صفحات قرار می‌دهند که می‌تواند خطرناک باشد.

۵. آیا استفاده از افزونه‌ای برای درک راحت‌تر اطلاعات استخراج شده از گواهینامه X.509 می‌تواند منجر به بروز مشکلات امنیتی شود؟

یکی از چالش‌های اصلی افزونه‌ها در مرورگرها، دسترسی به اطلاعات حساس است. افزونه‌ای که اطلاعات مربوط به گواهینامه‌های X.509 را استخراج می‌کند، به طور بالقوه به گواهینامه‌های SSL/TLS و اطلاعات رمزنگاری وب‌سایت‌ها دسترسی دارد. اگر این افزونه به درستی ایمن‌سازی نشده باشد، می‌تواند به عنوان یک درگاه برای حملات مهندسی اجتماعی یا افشای اطلاعات مورد سوءاستفاده قرار گیرد.

افزونه‌ها به طور پیش‌فرض می‌توانند ترافیک ورودی و خروجی مرورگر را کنترل کنند. یک افزونه مخرب می‌تواند با جعل اطلاعات گواهینامه یا تغییر داده‌ها در هنگام نمایش، کاربر را به صفحات فیشینگ هدایت کند. به ویژه اگر افزونه به طور مستقیم از APIهای مرورگر برای استخراج گواهینامه استفاده کند، می‌تواند باعث حملات مرد میانی (MITM) شود.

یکی دیگر از چالش‌ها این است که کاربران ممکن است از افزونه‌های غیررسمی یا ناشناس استفاده کنند. این افزونه‌ها می‌توانند به صورت مخفیانه اطلاعات استخراج شده را به سرورهای مخرب ارسال کنند. همچنین، اگر افزونه دارای مجوزهای بیش از حد باشد، امکان جمع‌آوری اطلاعات کاربر نیز وجود دارد.

که راهکارهایی برای جلوگیری از این خطرات امنیتی کانند کاهش سطح دسترسی افزونه و یا منبع باز کردن کد پروژه و استفاده از مرورگرهای ایمن مانند chrome و Firefox وجود دارد.

۶. آیا افزونه می‌تواند به گونه‌ای توسعه یابد که علاوه بر استخراج اطلاعات، یک ارزیابی اولیه از سطح ریسک امنیتی یک وب‌سایت ارائه دهد؟

افزونه‌ای که بتواند علاوه بر استخراج اطلاعات، ارزیابی ریسک امنیتی را نیز ارائه دهد، به کاربران کمک می‌کند تا با اطمینان بیشتری از وب‌سایت‌ها استفاده کنند. با استفاده از الگوریتم‌های پیشرفته و تحلیل دقیق گواهینامه‌ها، این افزونه می‌تواند به صورت خودکار پارامترهای کلیدی امنیتی را بررسی کرده و یک شاخص امنیتی (Security Score) ارائه دهد.

الگوریتم‌ها و شاخص‌های ارزیابی ریسک:

۱. ارزیابی کیفیت گواهینامه

- مدت زمان اعتبار (Validity Period): گواهینامه‌های کوتاه‌مدت امن‌تر هستند.
- سطح رمزنگاری (Encryption Level): استفاده از الگوریتم‌های امن مانند SHA-256.

- طول کلید عمومی: حداقل 2048 بیت برای RSA.

۲. تجزیه و تحلیل زنجیره اعتبار (Certificate Chain)

- صحت زنجیره اعتبار: بررسی اینکه زنجیره کامل باشد.
- ریشه معتبر (Trusted Root): اطمینان از اینکه گواهینامه به یک CA معتبر ختم شود.

۳. هشدار در صورت استفاده از الگوریتم‌های ضعیف

- الگوریتم‌هایی مانند MD5 یا SHA-1 باید به عنوان ناامن شناخته شوند.
- هشدار در صورت استفاده از کلیدهای کوتاه مانند 1024 بیت RSA.

۴. الگوریتم‌های تحلیل ریسک

- استفاده از مدل‌هایی مانند Random Forest یا Gradient Boosting برای شناسایی الگوهای مشکوک.
- تحلیل امتیاز ریسک (Risk Scoring): ترکیبی از عوامل امنیتی مختلف برای ایجاد یک نمره کلی امنیتی.