ShahnamehMap — سند ۴٫۶: مدل تهدیدات امنیتی

نسخه: ۱٫۰

تاریخ: ۲۰/۰۸/۱۴۰۳

تهیه‌کننده: مدیر فنی (CTO) با مشاوره امنیتی

وضعیت: محرمانه – دسترسی محدود

بازبینی: فصلی

---

۱. هدف و دامنه

این سند به شناسایی **تهدیدهای امنیتی احتمالی** علیه پلتفرم ShahnamehMap، دارایی‌های حیاتی**، و **کنترل‌های کاهش ریسک** می‌پردازد. هدف، **حفاظت از سرمایه** (داده‌ها، اعتبار برند، تداوم کسب‌وکار.) و **ایجاد اعتماد** در کاربران و شرکا است

ها، سرویس‌های APIدامنه:** کلیه اجزای سیستم شامل: فرانت‌اند وب، **
.بک‌اند، پایگاه‌های داده، زیرساخت کلاد، و فرآیندهای انسانی

---

۲. دارایی‌های حیاتی و حساس (Critical & Sensitive Assets)

دارایی	**حساسیت**	**دلیل حساسیت**	**مالک**
اطلاعات شخصی کاربران (PII)	بسیار بالا	ایمیل، نام نمایشی، هش رمز عبور. نقض = ریسک حقوقی (قانون حمایت از داده) و از دست رفتن اعتماد.	CTO
داده‌های مالی	بسیار بالا	توکن‌های پرداخت، تاریخچه اشتراک‌ها، جزئیات صورتحساب مدارس (B2B).	CTO
داده‌های محتوای کاربران (UGC)	بالا	کمپین‌ها، کاراکترهای ساخته‌شده. نقض = از دست رفتن دارایی فکری کاربران و آسیب به برند.	CCO

پایگاه داده دانش شاهنامه** | بالا | دارایی فکری اصلی شرکت. ** |
| CCO | .دستکاری = آسیب به اعتبار علمی و ارزش پیشنهادی

API بسیار بالا | کلیدهای | **(Secrets) کلیدهای دسترسی و اسرار** |
| .JWT ، رمزهای دیتابیس، کلیدهای(SendGridمثل درگاه پرداخت،)
CTO |

| CTO | .کد منبع** | بالا | دارایی فکری، امکان تزریق بک‌در** |

زیرساخت اجرایی** | بالا | سرورها، شبکه. از کارافتادگی = توقف ** |
| CTO | .کسب‌وکار

(Attacker Personas) پرسونای مهاجمان .۳ ##

مهاجم	**انگیزه**	**توانایی/منابع**	**اهداف محتمل**
سرگرمی، نشان دادن	**(Script Kiddie) مهاجم فرصت‌طلب**
مهارت.	استفاده از ابزارهای اتوماتیک برای شناسایی آسیب‌پذیری‌های
.سایت، سرقت اطلاعات کاربران ساده Deface	.شناخته شده

هکر رقبا یا مخالف فرهنگی** | اخلال، تخریب اعتماد، سرقت دارایی ** |
| .فکری. | مهارت فنی متوسط تا بالا، ممکن است سرمایه داشته باشد
| ., استخراج پایگاه داده شاهنامه، انتشار محتوای جعلی یا توهین‌آمیزDDoS

انتقام، سود مالی. | دسترسی | **(Insider) کاربر مخرب داخلی** |
داخلی به سیستم‌ها (توسعه‌دهنده ناراضی، کارمند اخراجی). | سرقت داده،
| .حذف داده، نصب بک‌در

هدف بلندمدت: جاسوسی، اخلال. | | **(APT) مهاجم سازمان‌یافته** |
منابع بسیار بالا، تخصص عمیق، صبر. | نفوذ پایدار به زیرساخت برای نظارت یا
| کنترل آینده. (کم احتمال، ولی باید در نظر گرفته شود)

کاربر سوءاستفاده‌گر** | دسترسی غیرمجاز به منابع، تخریب جامعه. | ** |
دسترسی به عنوان یک کاربر عادی. | تلاش برای دور زدن قوانین بازی
| .کاربران دیگر ト ン メ، ارسال محتوای نامناسب، هاراس(Cheating)

---

۴. سطح حمله و تهدیدهای اصلی (Attack Surface & Threats)

۴,۱. لایه کاربر/برنامه (Application Layer)

* **مرتبط OWASP Top 10:**

* **A01: Broken Access Control:** کاربری بتواند به داده کاربران دیگر یا پنل مدیریت دسترسی یابد.

* **A02: Cryptographic Failures:** انتقال یا ذخیره غیرایمن (مثلاً توکن‌ها در LocalStorage بدون HttpOnly) داده‌های حساس.

* **A03: Injection:** تزریق SQL در کوئری‌های دیتابیس، یا تزریق NoSQL در Neo4j.

* **A07: Identification & Authentication Failures:** مکانیزم‌های ضعیف بازیابی رمز عبور، عدم محدودیت تلاش برای ورود.

* **A08: Software & Data Integrity Failures:** استفاده از بسته‌های مخرب NPM (Supply Chain Attack).

۴,۲. لایه شبکه و زیرساخت (Network & Infrastructure Layer)

* **DDoS Attacks:** حملات حجمی برای از کار انداختن سرویس.

* **Misconfiguration of Cloud Services:** bucketهای S3/MinIO در دسترس عموم، پورت‌های مدیریتی باز.

در صورت استفاده نادرست از **Container Escape:** *
Docker/K8s.

۴,۳. لایه انسانی و فرآیند (Human & Process Layer)

* **مهندسی اجتماعی (Social Engineering):** فریب کارکنان برای افشای اسرار یا اعطای دسترسی.

* **عدم چرخش به موقع رمزها و کلیدها.**

* **عدم حذف دسترسی کارکنان اخراجی.**

---

۵. کنترل‌های امنیتی (Security Controls)

۵,۱. کنترل‌های پیشگیرانه (Preventive Controls)

* **احراز هویت و مجوز (AuthN/AuthZ):**

* **Keycloak** با **OAuth 2.0/ OIDC** و جریان **Authorization Code with PKCE**.

* **JWT** های کوتاه‌عمر** (۱۵ دقیقه) با قابلیت رفرش (Refresh Tokens در HttpOnly Secure Cookies).

* **اعتبارسنجی دقیق مجوز (RBAC)** در سطح API Gateway و هر سرویس.

* **محدودیت نرخ ورود (Rate Limiting)** و قفل‌کردن حساب پس از ۵ تلاش ناموفق.

* **محافظت در برابر تزریق:**

* **استفاده از ORM (Prisma)** با پارامترایزیشن خودکار برای PostgreSQL.

* **اعتبارسنجی ورودی (Input Validation)** سراسری با **Zod**.

* **اجرای کوئری‌های Neo4j** فقط از طریق توابع از پیش تعریف‌شده و پارامترایز شده**.

* **امنیت داده در حال انتقال و ذخیره:**

* **TLS 1.3** در همه جا (با گواهی Let's Encrypt یا معادل).

* **رمزنگاری داده‌های حساس در حال استراحت (Encryption at Rest)** برای دیتابیس‌ها و Object Storage.

* هش کردن رمزهای عبور با **Argon2id** یا **bcrypt** (قوی).

* **امنیت زیرساخت:**

* پیکربندی سخت‌گیرانه (Hardening) سرورها و کانتینرها.

* شبکه خصوصی مجازی (VPC)** برای ایزوله کردن سرویس‌ها.

* فایروال برنامه کاربردی (WAF)** در لبه (Edge) با استفاده از Cloudflare** برای فیلتر کردن ترافیک مخرب.

* منظم برای تصاویر Docker اسکن آسیب‌پذیری** و وابستگی‌ها با **Snyk**/**Trivy**.

۵,۲.** کنترل‌های کشف‌کننده (Detective Controls)**

* لاگینگ و مانیتورینگ جامع (SIEM):**

* ساختاربندی (Structured) همه لاگ‌ها با Correlation ID).

* ارسال لاگ‌های امنیتی (ورود ناموفق، تغییرات دسترسی، خطاهای احراز هویت) به یک سیستم** متمرکز SIEM مانند) **Wazuh** یا **Elastic SIEM**).

* قوانین هشدار (Alerting Rules)** برای شناسایی فعالیت مشکوک:

* تعداد بالای درخواست ۴۰۴ از یک IP.

* تلاش برای دسترسی به endpointهای مدیریتی.

* حجم غیرعادی داده از دیتابیس توسط یک کاربر دانلود.

* **نظارت بر یکپارچگی فایل (File Integrity Monitoring - FIM):** نظارت بر تغییرات فایل‌های کانفیگ حیاتی.

* **SAST & DAST:** اجرای **Static Application Security Testing** و **Dynamic Application Security Testing** در CI/CD (با **Snyk Code**) روی محیط Staging دوره‌ای.

۵,۳. کنترل‌های اصلاحی (Corrective/Responsive Controls)

* **پاسخ به حادثه (Incident Response Plan - IRP):** در بخش ۷ توضیح داده شده است.

* **پشتیبان‌گیری و بازیابی (Backup & Recovery):**

* پشتیبان روزانه **تام (Full)** + **افزایشی (Incremental)** از دیتابیس‌ها.

* **تست بازیابی (Restore Test)** ماهانه.

* پشتیبان‌گیری‌ها **رمزنگاری شده** و در **مکانی جداگانه** ذخیره می‌شوند.

* **Patch Management:** به‌روزرسانی منظم و سریع سیستم‌عامل، رانتایم‌ها و وابستگی‌های نرم‌افزاری.

---

۶. مدیریت اسرار و کلیدها (Secrets Management)

اصل: "اسرار هیچ‌وقت نباید در کد، ریپازیتوری یا **environmentهای plaintext ذخیره شوند."**

* ابزار: استفاده از یک **Secrets Manager** متمرکز مانند **AWS Secrets Manager**) یا سرویس مدیریت‌شده کلود **HashiCorp Vault**).

* **چرخه عمر:**

* همه اسرار (API رمز دیتابیس، کلید) به صورت خودکار تولید می‌شوند.

* چرخش منظم (هر ۹۰ روز یا پس از خروج هر کارمند)**

فقط به رمز Game مثلاً سرویس) **Policy** دسترسی مبتنی بر** *
.(دسترسی دارد Redis

محلی که `.env` در محیط توسعه:** استفاده از فایل‌های** *
هستند. استفاده از `.gitignore` نمی‌شوند و در commit **هرگز**
.**Direnv** ابزاری مثل

---

۷. برنامه پاسخ به حادثه (Incident Response Plan)

هدف: کاهش آسیب و بازیابی سریع.

مراحل IRP:

مرحله	**اقدامات**	**مسئول**
۱. آمادگی (Preparation)	تدوین IRP، آموزش تیم، تعیین تیم پاسخ (CTO، DevOps، PR).	CTO

• | **(Identification & Analysis) شناسایی و تحلیل ۲.** |
•
.تعیین دامنه و تاثیر •
.، لاگ‌هاSIEM جمع‌آوری شواهد از
| DevOps/تیم امنیت | .(P0-P3) طبقه‌بندی حادثه

• | **(Containment & Eradication) مهار و ریشه‌کنی ۳.** |
•
.(، غیرفعال کردن توکن/کاربرIP مسدود کردن) قطع دسترسی مهاجم
اعمال •
.حذف بک‌در/مالور •
.ایزوله کردن سیستم‌های آلوده
| CTO با نظارت DevOps | .پچ‌های امنیتی

بازیابی سیستم‌های تمیز از • | **(Recovery) بازیابی ۴.** |
تغییر همه رمزها و •
.از سرگیری سرویس‌ها با نظارت •
.پشتیبان
| DevOps | .کلیدهای مربوطه

تحلیل ریشه • | **(Post-Incident Activity) درس‌آموزی ۵.** |
مستندسازی درس‌ها و •
.(Root Cause Analysis - RCA)
اطلاع‌رسانی شفاف به کاربران ** •
.به‌روزرسانی کنترل‌ها
| کل تیم + CTO | .آسیب‌دیده** (در صورت نقض داده)

نکته کلیدی در اطلاع‌رسانی:** در صورت نقض داده کاربران، طبق **
قانون جرایم رایانه‌ای و عرف اخلاقی، در اسرع وقت و با شفافیت (بدون
پرده‌پوشی) به کاربران آسیب‌دیده اطلاع داده و راه‌کارهای محافظتی (تغییر
.رمز عبور) ارائه می‌شود

---

۸. انطباق (Compliance) و حسابرسی (Auditing)

* **خط‌مشی امنیت اطلاعات (ISP):** رعایت خط‌مشی امنیت اطلاعات داخلی شرکت.
* **حسابرسی:** فعال‌سازی **Audit Logging** برای تمامی عملیات حساس (ایجاد/حذف کاربر، تغییر دسترسی، دسترسی به داده‌های حساس). این لاگ‌ها **غیرقابل تغییر (Immutable)** و برای **حداقل ۱۰ سال** نگهداری می‌شوند.
* **تست نفوذ (Penetration Test):** انجام **سالیانه** توسط یک شرکت معتبر شخص ثالث.

---

۹. ماتریس ریسک امنیتی (پالایش‌شده)

تهدید** | **احتمال** | **اثر** | **ریسک** | **کنترل‌های ** |
| **موجود** | **اقدام اضافی

| --- | --- | --- | --- | --- | --- |

سرقت اطلاعات کاربران از دیتابیس** | متوسط | بحرانی | **بالا** | ** |
اجرای | .WAFرمزنگاری در حال انتقال و ذخیره، کنترل دسترسی دقیق،
| .**Database Activity Monitoring (DAM)**

علیه سرویس بازی** | متوسط | بحرانی | **بالا** | استفاده DDoS** |
تنظیم دقیق‌تر | .mitigatio با قابلیت CDN/WAF (Cloudflare) از
| .و داشتن پلن ارتباطی با ارائه‌دهنده اینترنت WAF قوانین

های قدیمی** | کم | بحرانی | API از طریق SQL تزریق** |
بازبینی ** | .Zod ، اعتبارسنجی باPrisma **متوسط** | استفاده از
| .(Raw Queries) دستی** کلیه کوئری‌های دست‌نویس

کاربر مخرب انتشار محتوای نامناسب** | بالا | متوسط | **متوسط** ** |
پایه** ML | سیستم گزارش کاربران، بررسی دستی. | پیاده‌سازی **مدل
| .برای فیلتر خودکار تصاویر/متن‌های توهین‌آمیز

مخرب** | متوسط | بالا | Dependency نفوذ از طریق** |
، قفل کردن نسخه‌ها. | **اجباری CI/CD در Snyk **متوسط** | اسکن
| .برای به‌روزرسانی‌های ماژور **Code Review کردن

---

۱۰. **نتیجه‌گیری: امنیت به عنوان یک قابلیت کسب‌وکار**

امنیت ضعیف تنها یک **مشکل فنی** نیست؛ یک **ریسک تجاری چندوجهی** است:

* **ریسک حقوقی و مالی:** جریمه‌های سنگین ناشی از نقض حریم خصوصی (مطابق قوانین داخلی و بین‌المللی).
* **ریسک برند و اعتماد:** از دست دادن کاربران و شریکان که غیرقابل جبران است.
* **ریسک توقف کسب‌وکار:** از کار افتادگی طولانی‌مدت سرویس و از دست دادن درآمد.

این مدل تهدید و کنترل‌های پیاده‌شده، **چارچوبی پیشگیرانه، نظارتی و واکنشی** ایجاد می‌کند تا ShahnamehMap بتواند ضمن نوآوری و رشد، **حافظ سرمایه‌های خود و اعتماد کاربرانش** باشد. این سند زنده است و با تغییر محصول، تهدیدات و درس‌های حوادث، به‌روزرسانی خواهد شد.