Security Threat Model

۴٫۶٫۱ **مقدمه**

`nobatnou.ir` سیستم نوبت‌دهی آنلاین به دلیل ماهیت حساس پزشکی و ذخیره‌سازی داده‌های شخصی کاربران (نام، شماره تلفن، سوابق نوبت‌گیری) در معرض تهدیدات امنیتی متنوعی قرار دارد. این سند به شناسایی، ارزیابی و ارائه کنترل‌های امنیتی برای کاهش ریسک‌های امنیتی می‌پردازد.

۴٫۶٫۲ **محدوده سیستم تحت بررسی**

- **صفحات وردپرسی با shortcode فرانت‌اند (پنل نوبت، ثبت‌نام، ورود، داشبورد).**
- **توابع PHP، Ajax handler بک‌اند، دیتابیس WordPress ها.**
- **جداول ذخیره‌سازی داده: `wp_nobat_system` و `wp_nobat_users`.**
- **درخواست‌های HTTP/HTTPS ارتباطات بین کلاینت و سرور.**
- **سیستم احراز هویت: session-based برای کاربران ثبت‌نام‌شده.**

۴/۶/۳ مهاجمان محتمل (Threat Actors)

مهاجم	انگیزه	سطح دسترسی اولیه	تهدیدات محتمل
۱. **کاربر عادی**	سودجویی شخصی، آزمایش سیستم	دسترسی به حساب خود	تلاش برای دسترسی به داده کاربران دیگر، سوءاستفاده از باگ‌های منطقی
۲. **کاربر خرابکار (Malicious User)**	تخریب، انتشار داده‌ها، اختلال در خدمات	دسترسی به حساب خود یا عمومی	حملات brute-force، اسکریپت‌نویسی XSS برای رزرو انبوه نوبت، تلاش برای
۳. **هکر بیرونی (External Hacker)**	دسترسی به داده‌ها، نفوذ به سرور، باج‌گیری	دسترسی از طریق اینترنت	اسکن آسیب‌پذیری، تزریق SQL، حملات DDoS، نفوذ به سرور
۴. **رقیب کسب‌وکار**	اختلال در خدمات، سرقت داده‌ها	دسترسی بیرونی	حملات منع سرویس (DoS)، استخراج داده از طریق باگ‌ها
۵. **کاربر داخلی (توسعه‌دهنده/ادمین)**	سوءاستفاده از دسترسی بالا	دسترسی به دیتابیس، کد، سرور	دسترسی غیرمجاز به داده‌ها، نصب بک‌دور، دستکاری در داده‌ها

۴/۶/۴ دارایی‌های حیاتی (Critical Assets)

| دارایی | حساسیت | توضیح |

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

| داده‌های شخصی کاربران** | بسیار بالا | نام کامل، شماره تلفن، سابقه نوبت‌ها .۱** |

| داده‌های نوبت‌ها** | بالا | موقعیت نوبت، زمان‌های ثبت، وضعیت نوبت‌ها .۲** |

| اعتبار کاربران** | بالا | شماره تلفن و رمز عبور هش‌شده .۳** |

دسترسی ادمین** | بسیار بالا | توانایی تغییر وضعیت نوبت‌ها، مشاهده همه داده‌ها .۴** |

| کد منبع سیستم** | متوسط | منطق کسب‌وکار، کوئری‌های دیتابیس .۵** |

تهدیدات شناسایی‌شده و کنترل‌های امنیتی ۴/۶/۵ ###

(SQL Injection) SQL تهدید ۱: تزریق ####

بدون (`phone_number`, `id`) Ajax بردار حمله:** پارامترهای ورودی** -
.کافی sanitize

:کنترل‌های پیاده‌شده -

.برای همه کوئری‌های دیتابیس `$wpdb->prepare()` استفاده از -

.`intval()` و `sanitize_text_field()` اعتبارسنجی ورودی‌ها با -

:کنترل‌های پیشنهادی -

- اجرای اسکن دوره‌ای با ابزارهای مثل WPScan.

- استفاده از WAF (Web Application Firewall) مانند Cloudflare.

۲ تهدید: XSS (Cross-Site Scripting)

- **حمله بردار HTML نمایش داده می‌شود.**

- **کنترل‌های پیاده‌شده:**

- استفاده از `esc_attr()` و `esc_html()` در خروجی‌های فرانت‌اند.

- **کنترل‌های پیشنهادی:**

- اضافه کردن `Content-Security-Policy` header.

- اعتبارسنجی سخت‌گیرانه‌تر ورودی‌ها با regex.

۳ تهدید: CSRF (Cross-Site Request Forgery)

- **حمله بردار فرم‌های بدون توکن در Ajax.**

- **کنترل‌های پیاده‌شده:**

- nonce (ضعف) در حال حاضر از استفاده نشده است.

- **کنترل‌های پیشنهادی:**

- افزودن `wp_nonce` به تمام درخواست‌های Ajax.

- بررسی `referer` header در سرور.

تهدید ۴: احراز هویت ضعیف** (Weak Authentication)**

- **بردار حمله:** brute-force روی صفحه ورود، ذخیره‌سازی رمز عبور ناامن.

- **کنترل‌های پیاده‌شده:**

- استفاده از `wp_hash_password()` و `wp_check_password()`.

- محدودیت طول و نوع رمز عبور (حداقل ۴ رقم).

- **کنترل‌های پیشنهادی:**

- پیاده‌سازی محدودیت تلاش ورود (rate limiting).

- افزودن احراز دو مرحله‌ای (۲FA) برای ادمین‌ها.

تهدید ۵: عدم کنترل دسترسی مناسب** (Broken Access Control)**

- **بردار حمله:** کاربران عادی بتوانند نوبت دیگران را لغو یا وضعیت را تغییر دهند.

- **کنترل‌های پیاده‌شده:**

- بررسی `$_SESSION['nobat_user']` قبل از عملیات حساس.

- تطبیق `phone_number` کاربر با نوبت در عملیات لغو.

- **کنترل‌های پیشنهادی:**

- پیاده‌سازی سیستم نقش‌ها (Role-Based Access Control) برای ادمین/کاربر.

- لاگ‌گیری از تمام اقدامات حساس.

تهدید ۶: افشای اطلاعات** (Information Disclosure)**

- **بردار حمله:** خطاهای PHP نمایش داده شود، مسیرهای فایل آشکار شود.

:کنترل‌های پیاده‌شده -

- تنظیم `display_errors = Off` در production.

:کنترل‌های پیشنهادی -

- استفاده از صفحات خطای سفارشی.

- جلوگیری از indexing دایرکتوری‌ها توسط ربات‌ها.

تهدید ۷: حمله DDoS/DoS

- **درخواست‌های انبوه به صفحه نوبت یا Ajax بردار حمله:**.

:کنترل‌های پیاده‌شده -

- استفاده از هاست با محدودیت پهنای باند مناسب.

:کنترل‌های پیشنهادی -

- استفاده از سرویس CDN/WAF مانند Cloudflare.

- تنظیم rate limiting در سطح وب‌سرور (nginx/Apache).

مدیریت کلیدها و رمزنگاری ۴/۶/۶

رمزنگاری داده‌ها در حالت انتقال (In Transit)

:کنترل‌های پیاده‌شده -

.برای تمام ارتباطات HTTPS (SSL/TLS) استفاده از -

.(HTTP Strict Transport Security) HSTS تنظیم -

(At Rest) رمزنگاری داده‌ها در حالت ذخیره ####

:کنترل‌های پیاده‌شده -

هش (bcrypt با استفاده از) `wp_hash_password` رمز عبور کاربران با -
.می‌شود

:کنترل‌های پیشنهادی -

.AES-256 رمزنگاری ستون‌های حساس (مانند شماره تلفن) در دیتابیس با -

HashiCorp یا AWS KMS مثل) مدیریت کلیدهای رمزنگاری در محیط امن -
.(Vault

(Key Management) مدیریت کلیدها ####

.، کلیدهای دیتابیس (در صورت رمزنگاری)SSL کلیدهای موجود:** کلید** -

:سیاست‌ها -

.هر ۹۰ روز SSL چرخش دوره‌ای کلید -

.ذخیره کلیدها در محیط جدا از کد منبع -

برای اطلاعات (environment variables) استفاده از متغیرهای محیطی -
.حساس

۴٫۶٫۷ کنترل دسترسی (Access Control)

دسترسی به کد و سرور

- **دسترسی:** توسعه‌دهندگان با کلید عمومی (غیر از رمز عبور) SSH.

- **دسترسی به وردپرس با نقش ادمین‌ها:** دسترسی محدود به افراد `administrator` ضروری.

دسترسی به دیتابیس

- **دسترسی:** کاربر دیتابیس (`SELECT`, `INSERT`, `UPDATE`) اکانت جداگانه با حداقل دسترسی روی جداول مربوطه.

کنترل دسترسی در سطح اپلیکیشن

- **دسترسی:** کاربران عادی فقط به نوبت‌های خود دسترسی دارند.

- **دسترسی:** ادمین‌ها دسترسی به همه نوبت‌ها از طریق پنل مدیریت (نیاز به احراز هویت مجدد).

۴٫۶٫۸ لاگ‌گیری و مانیتورینگ

لاگ‌های امنیتی جمع‌آوری‌شده ####

| رویداد | محل لاگ | اطلاعات ثبت‌شده |

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

| ورود موفق/ناموفق | فایل سرور + دیتابیس | زمان، IP، شماره تلفن، user agent |

| ثبت نوبت | دیتابیس | زمان، نام بیمار، شماره تلفن، موقعیت نوبت |

| تغییر وضعیت نوبت | دیتابیس | زمان، کاربر عامل، شناسه نوبت، وضعیت جدید |

| لغو نوبت | دیتابیس | زمان، کاربر عامل، شناسه نوبت |

| خطاهای امنیتی (مثلاً SQLi تلاش) | فایل سرور | زمان، IP، payload، endpoint |

ابزارهای مانیتورینگ ####

- **سرور:** UptimeRobot برای مانیتورینگ uptime.

- **اپلیکیشن:** Sentry برای ردیابی خطاهای PHP/JS.

- **ترافیک:** Google Analytics + Cloudflare Analytics.

- **لاگ‌های امنیتی:** Fail2ban برای مسدودسازی IPهای مخرب.

هشدارهای امنیتی (Security Alerts) ####

- ارسال ایمیل به تیم فنی در صورت:

- بیش از ۵ تلاش ورود ناموفق از یک IP در ۵ دقیقه.

- تغییر در فایل‌های هسته وردپرس.

- خطاهای دیتابیس متعدد.

۴/۶/۹** پاسخ به حوادث امنیتی (Incident Response Plan)**

مراحل پاسخ به حادثه**

۱. شناسایی (Identification):

- دریافت هشدار از سیستم مانیتورینگ یا گزارش کاربر.

- تأیید حادثه (مثلاً نفوذ، داده‌نشتی).

۲. مهار (Containment):

- مسدودسازی IP حمله‌کننده در فایروال.

- قطع دسترسی کاربر آلوده (در صورت لو رفتن رمز عبور).

- گرفتن backup از لاگ‌ها و حالت فعلی سیستم.

۳. ریشه‌یابی (Eradication):

- بررسی لاگ‌ها برای شناسایی بردار حمله.

- رفع آسیب‌پذیری (مثلاً patch کد، به‌روزرسانی پلاگین).

- اسکن کامل سیستم برای backdoor یا malware.

۴. بازیابی (Recovery):

- بازگردانی سیستم از backup (در صورت نیاز) سالم.

- تغییر کلیدها و رمزهای عبور مربوطه.

- ازسرگیری سرویس پس از تست امنیتی.

۵. بررسی پس از حادثه (Post-Incident Review):

- مستندسازی کامل حادثه و اقدامات انجام‌شده.

- به‌روزرسانی مدل تهدیدات و کنترل‌های امنیتی.

- آموزش تیم در صورت نیاز.

تماس‌های اضطراری ####

- **مسئول هماهنگی پاسخ:** مدیر فنی.

- **برای مسدودسازی در سطح شبکه:** پشتیبانی هاست.

- **در صورت افشای داده‌های کاربران:** تیم حقوقی.

ارزیابی ریسک (Risk Assessment) ### **۴/۶/۱۰**

| ریسک | احتمال | تاثیر | سطح ریسک | اقدام کاهش |

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

، اسکن WAFمتوسط | بسیار بالا | بالا | بررسی کد، | **(SQLi) نفوذ به دیتابیس** |
| دوره‌ای

سرقت داده کاربران** | متوسط | بسیار بالا | بالا | رمزنگاری داده‌ها، کنترل ** |
| دسترسی دقیق

rate، CDN | پایین | بالا | متوسط | **(DoS) اختلال در سرویس** |
| ، هاست مقیاس‌پذیرlimiting

integrity دستکاری در نوبت‌ها** | بالا | متوسط | متوسط | لاگ‌گیری، بررسی** |
| داده‌ها

| افشای اطلاعات خطاها** | پایین | پایین | پایین | غیرفعال کردن نمایش خطاها** |

(Security Roadmap) برنامه اجرایی امنیتی ۴/۶/۱۱ ###

فاز ۱ (فوری - ۱ ماه) ####

- افزودن nonce به تمام Ajax requestها.

- پیاده‌سازی rate limiting برای صفحه ورود.

- تنظیم CSP header.

فاز ۲ (کوتاه‌مدت - ۳ ماه) ####

- رمزنگاری ستون‌های حساس در دیتابیس.

- پیاده‌سازی سیستم RBAC برای ادمین‌ها.

- نصب و پیکربندی WAF.

فاز ۳ (میان‌مدت - ۶ ماه) ####

- احراز دو مرحله‌ای برای ادمین‌ها.

- اجرای تست نفوذ دوره‌ای توسط 第三方.

- راه‌اندازی سیستم SIEM برای جمع‌آوری متمرکز لاگ‌ها.

نتیجه‌گیری ۴۰۶/۱۲ ###

با در نظر گرفتن تهدیدات محتمل و پیاده‌سازی `nobatnou.ir` امنیت سیستم کنترل‌های لایه‌ای (دفاع در عمق) طراحی شده است. با این حال، امنیت یک فرآیند مستمر است و نیازمند بازبینی دوره‌ی مدل تهدیدات، به‌روزرسانی کنترل‌ها و آموزش تیم است. این سند به‌عنوان راهنمای امنیتی جاری و آتی پروژه عمل خواهد کرد.