A7 - Proteção insuficiente contra ataques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Agentes de Ameaça | Vetores de Ataque | Vulnerabilidades de Segurança | | Impactos Técnicos | Impactos no Négocio |
| Especifico da Aplicação | Exploração  FACIL | Prevalência  COMMUM | Detecção  FACIL | Impacto  MODERADO | Especifico Negocio / Aplicação |
| Considere uma pessoa com acesso à rede que pode enviar um pedido a sua aplicação. A sua aplicação é capaz de detectar e reagir contra ataques manuais ou automatizados ? | Atacantes, utilizadores autorizado o não, enviam ataques. A aplicação o a API é capaz de detectar ao ataque ? E capaz de impedir ataques conhecidas ? | Aplicações e APIs são sempre atacados. A maioria das aplicações e dos APIs pode detectar dados não fiavieis, mas simplesmente a rejeita, deixando o atacante continuar o ataque. Tais ataques mostram que um utilizador comprometido o mal-intentianado tentam explorar vulnerabilidades. Detectar e bloquear ataques manuais ou automatizados é uma das maiores soluções para acrescentar a segurança. Com que rapidez você pode corrigir uma vulnerabilidade crítica que acabou de descobrir? | | A maioria dos ataques bem sucedidos começa com a sondagem de vulnerabilidades. Permitir tais sondas para continuar pode aumentar a probabilidade de exploração bem sucedida a 100%. Não instalando rapidamente as correções ajuda os atacantes. | Considere o impacto de ter uma proteção insuficiente contra ataques para o seu negocio. Ataques bem sucedidas não podem ser impedidas, não vão ser descobertas durante um longo prazo e podem expandir muito alem da exploração inicial. |

**Estou vulnerável?**

Detectar, reagir e bloquear ataques tornam as aplicações dramaticamente mais difíceis de explorar, mas quase nenhuma aplicação ou APIs tem este tipo de proteção. Vulnerabilidades críticas ambos num código personalizado e componentes são descobertas frequentemente, mas varias vezes as organizações levam semanas ou mesmo meses para lançar novas defesas.

E muito óbvio de ver se o processo de detecção e de resposta esta posto. Basta tentar ataques manuais ou lançar um scanner contra a sua aplicação. A aplicação o a API deve identificar os ataques, bloquear quaisquer ataques viáveis e fornecer detalhes sobre o atacante e as características do ataque. Se você não conseguir rapidamente distribuir correções virtuais e / ou reais quando uma vulnerabilidade crítica for descoberta, você ficará exposto ao ataque.

Certifique-se de compreender que tipos de ataques são cobertos pela proteção contra ataques. É apenas XSS e SQL Injection? Você pode usar tecnologias como [WAFs](https://www.owasp.org/index.php/Web_Application_Firewall), RASP e [OWASP App Sensor](https://www.owasp.org/index.php/OWASP_AppSensor_Project) para detectar ou bloquear ataques, e/ou virtualmente implantar todas as novas atualizações.

**Com faço para evitar ?**

Tem três objetivos principais para ter uma proteção suficiente contra ataques:

1. **Deteção de ataques.** Ocorreu alguma coisa que é impossível de ser causado por utilizadores legítimos (por exemplo, uma entrada que um cliente legitimo não pode fazer) ? A aplicação esta sendo usado de uma forma que um utilizador comum nunca faria (por exemplo, tempo demasiado elevado, entrada atípica, padrões de uso incomuns, solicitações repetidas) ?
2. **Reagir os ataques.** Registos de eventos e notificações são fundamentais para uma resposta oportuna. Decida se deseja bloquear automaticamente solicitações, endereços IP ou intervalos de IP. Considere a possibilidade de desabilitar ou monitorar contas de utilizador com comportamento estranho.
3. **Lançar correções frequentes.** Se o seu processo e desenvolvimento de software não é capaz de instalar correções criticas no mesmo dia, aplica correções virtuais que analisam o tráfego HTTP, o fluxo de dados, e / ou a execução de código e impede vulnerabilidades de ser exploradas.

**Exemplo de Cenário de Ataque**

Cenário #1: Um atacante usa uma ferramenta automatizada como [OWASP ZAP](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project) ou [SQLMap](http://sqlmap.org/) para descobrir vulnerabilidades e possivelmente explora-las.

Deteção de ataques devem reconhecer que a aplicação esta visada com pedidos não usais e muita frequência. Scans automatizados devem ser simples de identificar comparando com um tráfego normal.

Cenário #2: Um atacante humano qualificado examina cuidadosamente possíveis vulnerabilidades, eventualmente encontrando uma falha obscura.

Embora mais difícil de detectar, esse ataque ainda envolve solicitações que um utilizador normal nunca enviaria, como dados não autorizados pela interface. Acompanhar este atacante pode exigir a construção de um caso ao longo do tempo que demonstra intenção maliciosa.

Cenário #3: O atacante começa a explorar uma vulnerabilidade na sua aplicação que sua proteção de ataque atual não consegue bloquear.

Com que rapidez você pode implantar uma correção real ou virtual para bloquear a exploração continua desta vulnerabilidade ?

**Referências**

OWASP

[OWASP Article on Intrusion Detection](https://www.owasp.org/index.php/Intrusion_Detection)

[OWASP App Sensor](https://www.owasp.org/index.php/OWASP_AppSensor_Project)

[OWASP Automated Threats Project](https://www.owasp.org/index.php/OWASP_Automated_Threats_to_Web_Applications)

[OWASP Credential Stuffing Cheat Sheet](https://www.owasp.org/index.php/Credential_Stuffing_Prevention_Cheat_Sheet)

[OWASP Virtual Patching Cheat Sheet](https://www.owasp.org/index.php/Virtual_Patching_Cheat_Sheet)

[OWASP Mod Security Core Ruleset](https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project)

External

[WASC Article on Insufficient Anti-automation](http://projects.webappsec.org/w/page/13246938/Insufficient%20Anti-automation)

[CWE Entry 778 – Insufficient Logging](http://cwe.mitre.org/data/definitions/778.html)

[CWE Entry 799 – Improper Control of Interaction Frequency](http://cwe.mitre.org/data/definitions/799.html)