Introduction:

Cyber insurers are placing greater emphasis on API security during the underwriting process. Vulnerabilities in your API security can result in increased premiums, restricted coverage, or outright application denial. Utilize this checklist to assess your API security practices and pinpoint possible shortcomings that underwriters might scrutinize. Responding with 'No' or 'Partial' highlights areas that need attention to enhance your insurability.

Section	Item #	Question	Response (Yes/No/Partial/Planned)	Notes / Action Items
1. API Visibility & Inventory *(Why: Insurers need assurance you understand your complete attack surface. Unknown APIs = unmanaged risk.)*	1.1	Do we maintain a complete, accurate inventory of all external-facing APIs?		
	1.2	Does our inventory include internal (East-West) APIs?		
	1.3	Does our inventory include third-party APIs consumed by our applications?		
	1.4	Do we know which APIs handle or provide access to sensitive data (PII, PHI, PCI, etc.)?		

	1.5	Do we have an automated process to discover new or undocumented ('shadow') APIs?		
	1.6	Do we have a process to identify and decommission unused ('zombie') APIs?		
2. API Design, Testing & Posture Governance *(Why: Demonstrates proactive vulnerability management & adherence to best practices, reducing exploitation likelihood.)*	2.1	Are APIs designed following security best practices (e.g., considering OWASP API Top 10)?		
	2.2	Do we regularly perform security testing (SAST, DAST, IAST, Pen Test) specifically on our APIs?		
	2.3	Is API security integrated into our Software Development Lifecycle (SDLC) / DevSecOps process?		
	2.4	Do we enforce strong, modern authentication mechanisms for all APIs?		
	2.5	Do we implement granular authorization (e.g., Role-Based Access Control, Object-Level Authorization) based on the principle of least privilege?		

	2.6	Are appropriate rate limiting, throttling, and input validation controls implemented for APIs?		
	2.7	Are security configurations for API gateways, servers, and related infrastructure hardened and regularly reviewed?		
	2.8	For APIs handling regulated data, do controls meet relevant compliance requirements (PCI-DSS, HIPAA, GDPR, etc.)?		
3. API Threat Protection & Monitoring *(Why: Shows capability to detect/respond to active attacks, minimizing impact of breaches/claims.)*	3.1	Do we have dedicated runtime protection specifically for API traffic beyond legacy API Gateways/WAFs?		
	3.2	Do we monitor API traffic for anomalous behavior indicative of attacks (e.g., BOLA, credential stuffing, data scraping)?		
	3.3	Can our systems detect and block sophisticated attacks targeting API logic?		
	3.4	Do we have alerting mechanisms in place for critical API security events?		

	3.5	Do our incident response plans specifically address API security incidents?		
	3.6	Are relevant API security logs collected, correlated, and retained for analysis?		
4. Documentation & Process *(Why: Ability to prove controls/processes exist is crucial during underwriting/audits.)*	4.1	Can we readily provide documentation/evidence of our API inventory to insurers?		
	4.2	Can we readily provide documentation/evidence of our API testing procedures and findings?		
	4.3	Can we readily provide documentation/evidence of our API runtime security controls?		
	4.4	Is responsibility for API security clearly defined within our organization?		

Interpreting Your Results & Next Steps:

- Count the number of 'No' and 'Partial' answers. Each represents a potential gap in your API security posture that could raise concerns for cyber insurers.
- Prioritize addressing gaps in areas with the most 'No' answers, particularly concerning sensitive data handling, authentication/authorization, and runtime protection.
- Develop an action plan to implement missing controls or improve existing ones.

Strengthen Your API Security Posture:

Addressing these areas often requires specialized tools beyond traditional security solutions. Consider exploring dedicated API security platforms, such as Salt Security, which provide continuous discovery, posture governance, and runtime threat protection.