APIs COME UNDER ATTACK

Application protocol interfaces (APIs) are a double-edged sword for modern applications such as mobile apps, IoT apps and third-party services embedded into existing applications. They simplify architecture and delivery but introduce a wide range of risks and vulnerabilities. Unfortunately, API vulnerabilities still do not get the required visibility. All of the risks that affect web applications also affect web services, and yet traditional application security assessment tools such as Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST) either don't work well with APIs or are simply irrelevant to them.

APIs will be at the heart of many upcoming technological capabilities, but protecting them will be one of the gravest concerns of cyber-security professionals for years to come. Based on research from Radware's *2017-2018 Global Application & Network Security Report*, here is a list of concerns for APIs - many of which will be attacked in 2018 and beyond.

- **TLS** is required to secure the communications between the client and APIs for transport confidentiality and integrity of data in transit.
- **TCP Termination** for network evasion attacks detection where IP fragmentation is applied.
- **HTTP protocol parsing** and enforcement of HTTP RFC protects against various HTTP attacks such as NULL byte injection, encoded attacks, HRS attacks, content-type mismatch, etc.
- **Traffic normalization** for evasion attacks detection. Encoded attacks can easily bypass security solutions.

- **Message size policy** enforcement on HTTP message, body, headers and JSON/XML element sizes secures the application against buffer overflow attacks, resource exhaustion and other availability attacks on API infrastructure.

- **Access control** policy management with:
  - **IP-based** and **geo location** restrictions when relevant
  - **Access restriction to particular APIs** where, for example, some APIs should be exposed for public access while others are just for internal use.
  - **Access restrictions to specific HTTP methods** where the set of operations allowed for certain users are prohibited for other users or sources. (For example, a user can generate a license but cannot delete the license once generated.)

- **Strong typing** and a **positive security model** provide tight protection to the API infrastructure. It will be impossible to generate most of the attacks if, for instance, the only allowed value type in the JSON element is an integer with the value range of 1 – 100.
- **XML/JSON validity check and schema validation** is an extremely important security protection. Types, value ranges, sizes and order of XML elements must be configurable.
- **Rate-based protection** per application or per API is an important protection against service abuse (for informational APIs), brute force attacks and DoS attacks.
- **XSS** protection should be based on rules and signatures of known attack patterns.
- **SQL and no-SQL injection** protections can be achieved by sanitizing and validating user inputs and via rule-based attack detection.
- **Session management** can be used to protect the API key, which is posted as a body argument or in the cookie.
- **Data leak protection** is essential to making sure error messages and sensitive information is not leaking out to the potential attacker.
- **DDoS protection** is key to preventing and mitigating a wide variety of DDoS attack techniques that may exploit API vulnerabilities.

DOWNLOAD THE *2017-2018 GLOBAL APPLICATION & NETWORK SECURITY REPORT* TO LEARN MORE

LEARN MORE AT DDOS WARRIORS

To know more about today's attack vector landscape, understand the business impact of cyber-attacks or learn more about emerging attack types and tools visit DDoSWarriors.com. Created by Radware's Emergency Response Team (ERT), it is the ultimate resource for everything security professionals need to know about DDoS attacks and cyber security.

This document is provided for information purposes only. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law. Radware specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionalities, services, or processes described herein are subject to change without notice.