AppCheck

ACCURACY IS EVERYTHING

E: info@appcheck-ng.com
W: www.appcheck-ng.com
T: 0113 887 8380

How does AppCheck work?

An all-encompassing approach to vulnerability scanning

- OSINT - Intelligence Gathering
- Gateway Layer - Scanning the gateway, intermediate networking components and remote access/management components.
- Network delivery and presentation layer - Scanning proxies, caching servers, load balancers, and presentation servers
- Application Framework, CMS, and Hosting layer - Application frameworks, Content Management Systems.
- Application code / Input processing (DAST) - Identifying vulnerable code through crawling and parameter testing.
- Cloud and Third-Party Trust layer - Third party trusts

OSINT / Intelligence Gathering

Using multiple Open-Source Intelligence (OSINT) to gather information that can be seeded into the assessment process.

During the discovery phase, the scanner consults multiple open-source intelligence databases to learn as much about the target system as possible. For example, host names registered to the target IP address, web components indexed by search engines, and historical network data. Data that is in scope for the scan is then seeded into the scan configuration.

Gateway Layer

Scanning systems such as firewalls, remote access, and management solutions to identify security flaws.

AppCheck uses multiple dedicated infrastructure scanners to identify vulnerabilities on each accessible network device. The scan begins by port scanning each IP address within the scope to identify accessible services. Each identified service is then probed for vulnerabilities using tens of thousands of checks.

Network Delivery and Presentation Layer

Identifying vulnerabilities within hosting infrastructure used to manage and optimise network traffic to web application servers.

AppCheck combines infrastructure scanning with web application build review check to analyse the flow of data from the scanning node to the target system. Identified systems are checked for known vulnerabilities using a regularly updated vulnerability database that combines well know sources such as the National Vulnerability Database (NVD) with our own internally maintained vulnerability feed.

Application Framework, CMS, and Hosting Layer

Identifying vulnerabilities within Application Frameworks such as ASP .NET, PHP, NodeJS, Java, Apache Tomcat/Struts, Spring, WebLogic, Django, Ruby on Rails and many more.

The AppCheck Web Application scanning engine includes dedicated scanners for a wide range of popular CMS systems and Application Servers and Frameworks. Each scanner is integrated with the Dynamic Security Testing engine so that it can be deployed in the correct way as applicable systems are identified during web crawling and discovery.

Checks for known vulnerabilities, such as those with a CVE identifier, are deployed in the same way and are regularly updated based via AppCheck's own vulnerability database and several community driven vulnerability feeds (updated daily).

By integrating platform checks within the web application scanning engine, components enumerated during this phase can be passed forward into other scanning layers for further scanning. For example, CMS plugins enumerated during forced browsing checks can then be passed to the DAST scanning engine to discover previously undisclosed vulnerabilities (0day).

Application Code / Input Processing (DAST)

Detecting security flaws within application code through Dynamic Application Security Testing (DAST).

For each URL configured with the scan, AppCheck performs online reconnaissance to gather information pertaining to the site that is publicly available in search engines and other online indexing services. Next AppCheck will map out the application using a sophisticated crawling engine. The crawler combines traditional web scraping with a browser-based crawler which implements artificial intelligence to mimic typical application user behaviour.

The "Mapped Attack Surface" enumerated during the initial phases of the scan is then subject to methodical security testing. Typically, the assessment process works by taking each user supplied data component, such as a form field of query string parameter, then modifies it to include a specific test case before submitting it to the server. Based on the applications response, further test cases are then submitted through the same method to confirm the vulnerability.

Cloud and Third-Party Trust Layer

Identify third-party components and trust relationships and identify vulnerabilities that arise through the use of vulnerable components and Cloud Service configuration vulnerabilities.

AppCheck audits all third-party trust relationships for subdomain takeover and related flaws.

AppCheck Identifies known vulnerabilities within deployed JavaScript libraries.

AppCheck assesses Amazon Simple Storage Service (S3) buckets for misconfigurations. This includes insecure permissions and bucket takeover vulnerabilities.

Some vulnerabilities such as Server-Side Request Forgery (SSRF) can have a greater impact when hosted within a cloud environment. AppCheck includes several cloud specific checks to detect and safely exploit vulnerabilities in cloud systems.

AppCheck identifies JavaScript malware, Card Skimmers and Crypto Mining software. It will also provide a domain report of third-party software including domain age, geolocation and susceptibility to domain takeover.

Learn more about AppCheck's approach to vulnerability scanning on our website:

appcheck-ng.com/our-approach/