Nemasis - DAST

Dynamic Application Security Testing

Web Applications

In the era of globalization, web-applications are turning to become a part of the IT assets of an organization. The open-source or third-party developed web-based applications are enabling organizations to ensure ease in conducting their business operations. Besides, various organizations are migrating from legacy systems to web-based applications, which provide them with centralized data warehousing capabilities. The digital threats have developed from attacking FTP, Telnet/SSH, and mail servers as a large number of services are exposed to the Internet have increased significantly, however, in recent years, web applications play a critical role in half of the breaches that happen around the world. Web applications have become the simplistic route for the hacker as they prove to be least resistant to infiltration and gain access to the internal network/resources of an organization.

Network scanners and assessment tools are responsible for validating and verifying the presence of vulnerabilities within the network and the accessible assets. However, in the case of a web application, it is the business logic which needs to be tested, since any Network Vulnerability Scanner would scan the web-servers and not the content which is being pulled away. The most common victims of web application breaches are Content Management Systems (CMS) like WordPress, Drupal, etc., Database Administration tools like phpMyAdmin, and SaaS applications. Web applications are developed using different frameworks and methodologies, which are designed to ensure faster delivery of the content to ensure business continuity.

Web Application Scanner/Application Security Audit

Web application scanner enables organizations to scan applications for vulnerabilities which may have been left exposed during the development cycle and it is an essential part of Enterprise Security Testing. Web applications are one of the most vulnerable facets of enterprise security – more than 50% of all successful data leaks and breaches comprises of web apps. IT Administrators and Developers would probably identify vulnerabilities inside the hosted applications like Cross-Site-Scripting (XSS) or SQL Injection (SQLi) as well as backdoors and other threats that hackers may exploit to attack an organization.

Dynamic Application Security Testing (DAST) is a black-box security testing technique in which an application is tested in its operating state and tries to hack it just like a hacker would. A DAST test not only looks for a wide range of vulnerabilities, including input/output validation issues that could leave an application vulnerable but also facilitates the organization to follow the industry-based compliances.

Security threats posed by web applications

There are various drawbacks when it comes to relying on web applications for business processes. Web applications pose a significant threat to the digital environment of any organization with a myriad of vulnerabilities and attack vectors. The most important thing all organizations will have to address and guard themselves against the presence of software vulnerabilities and threats to web applications. While there is no 100% assurance for safety, there are some steps one can undertake to avoid wreak havoc.

The OWASP Top 10 is a report document of security concerns for web application security. It represents a wide-ranging consent about the most critical security risks to web applications; however, an application may be vulnerable to a variety of attacks which may never make it to the OWASP Top 10.

Here are some common threats to look out for and avoid while using web applications:

Injection Attacks: Web application may contain defects which enable a hacker to inject code into the application or modify the requests/parameters which have not been disinfected. When an application requires more data, the chances for injection attacks are more. Some names of the injection attacks are SQL injection (SQLi), cross-site scripting (XSS), stored XSS Remote File Inclusion (RFI) and Local File Inclusion (LFI) which are the result of incorrect coding practices and are difficult to detect amid the development stage.

Broken Authentication: When authentication functions related to the web application are not executed correctly, it enables hackers to compromise passwords or session ID's, data access URIs are accessible without authentication, or to exploit other implementation flaws using other users credentials to gain access to restricted areas or may leave the data disclosed.

Insecure Direct Object References: Web applications may provide a feature to download a file, however, the hacker is able to download any file from inside the system is one of the examples of Insecure Direct Object Reference. There have been instances wherein hackers have accessed sensitive data files/password files through this vulnerability.

Security misconfiguration: Hackers depend on system errors & debug data to increase the understanding of a system, those are introduced due to a misconfiguration of the server. However, executing an application with known vulnerabilities is also a part of the Insecure Server Configuration. These attacks depend on the hacker's knowledge of vulnerabilities which have been made publicly accessible, patches have been made available by the developers, and the server administrators have failed to implement the patches.

Mirai botnet exploited the Security Misconfiguration of the IOT devices and had successfully affected poorly protected Internet devices (IoT) by using telnet to exploit default username and password. Also, it has been one of the most successful botnets in the history of IT Security.

Summary of Features – Nemasis DAST

Web-Applications are prone to numerous vulnerabilities which can be harder to detect by manual testing. Nemasis-DAST uses Passive Scanning and Attack /Active Scanning modes to identify and exploit these vulnerabilities.

Passive Mode Scanning

- Missing headers related to
  - CSRF Tokens
  - Cache-Control
  - Content-Type
  - Content Security Policy
- Analyzes Cookies, Cookie Poisoning
- Information Disclosure through
  - Error Codes
  - Referrer Header
  - Comments
  - X-Debug-Token
- Private IP disclosure
- Reverse Tab-nabbing
- WSDL File Scanning
- Cross-Domain Misconfiguration
- PII (Personal Identifiable Information)
- URL Rewrites – Session ID

Attack/Active Mode Scanning

- Remote File Include, Server Side Include Attacks
- Remote OS Command Injection, Remote Code Execution
- Directory Browsing/ Traversal
- CRLF injection, XPath Injection, SQL Injection
- Parameter Tampering
- Cross Site Scripting - Persistent, DOM based
- Open SSL Vulnerability (HeartBleed)
- Session Fixation
- Backup File Disclosure
- SOAP Attacks – Action Spoofing, XML Injection
- ELMAH (Error Logging Modules and Handlers) Information Leak

Features – Nemasis DAST

Spider/Crawling: Nemasis-DAST's spider is a tool that is used to automatically discover new resources (URLs) on a specific site. It starts with a list of URLs to visit, called the seeds, which depends on how the Spider is started. It then visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit and the process continues recursively as long as new resources are found. Amid the processing of an URL, it makes a request to fetch the resources and then parses the responses, and recognizing hyperlinks.

Passive Scanning/Non-Intrusive Scan: Nemasis-DAST passively scans all HTTP messages (requests and responses) sent to the web applications and is safe to use since it does not change the requests or responses. This is performed in a background thread to guarantee that it doesn't back off the analysis of an application. Passive scanning can also be used for automatically adding tags and raising alerts for potential issues which are provided by default.

Active/Attack Scanning / Intrusive Scan: Active scanning endeavors to discover potential vulnerabilities by using known attacks against the selected targets. It is an attack on those targets which you should NOT use on web applications that you do not own. It can discover vulnerabilities like broken access control; will not be found by any active or automated vulnerability scanning.

AJAX Scanning: It is a security scanner used to evaluate the security of AJAX-enabled applications. By detecting the specific AJAX frameworks in use, Nemasis-DAST is able to better formulate test requests and identify potential vulnerabilities.

Compliance and Configuration Assessment: Nemasis allows fast-track the compliance assessments of web applications and infrastructure according to industry standard and best practices such as Payment Card Industry (PCI), General Data Protection Regulation (GDPR), OWASP 2013, OWASP 2017, SANS Top 25, and many more. With these reports, users can identify the security gaps in the web applications.

Data Backup and Restore: Nemasis DAST Manage Instance feature helps you to create a backup of all types of data stored as a result of the scans and generated reports in case of any disaster or failure of system. This will help the organizations to retain the data in such cases by exporting and importing them in a ZIP format. The ZIP file includes all data such as, Configuration Files, License Information, Generated Reports, Scan Logs, and more.

Two Factor Authentication (2FA): Nemasis 2FA feature provides an extra layer of security to the Nemasis instance with an additional authentication layer. To enable this feature the user must be an Administrator, the user get time-based one-time password (TOTP) through Authenticator app. Once Administrator enables it, he/she will have provide TOTP for every login. Administrator can enable this feature for other users as well through Nemasis console.

Services: Nemasis DAST includes features that allow to perform an audit scan and provides with analysis, corrective suggestions, and solutions for various services such as, WHOIS, SEO Analytics, Domain security posture, Malware Check, MongoDB security audit, SSL security configuration, and Domain BlackList status. This helps organizations to not only overcome the vulnerabilities in the web applications but also the loopholes in the organizations that would delay the businesses to achieve its goals.

Administration: Nemasis DAST allows you to configure and manage different users with different sets of roles and permissions. The administrator account is created by default during installation and can create and manage other users. Nemasis allows to create role-based users thereby providing segregation of duties in a DAST program of organizations. Integration to LDAP is also supported which will help in being in-line with access control policies of DAST-based or define as per the organization's security policies.

Reports: Nemasis DAST provides detailed reports of all the vulnerabilities found in the web applications, which includes WASC ID, CWE, and many more. The reports generated are real-time and is in HTML format. Nemasis DAST provides three types of reports, namely, Nemasis-DAST Report, OWASP 2017, and OWASP 2013 that includes the recommended remediation for the vulnerabilities found. User can customize the header of the report through Customize Header feature according to the organization's requirement for a header format.

Scan Policy Management: This feature of Nemasis-DAST enables you to manage the scan policies that define the rules that are run while performing a scan. You can add the number of scan policies as you like, and choose which policy should be executed while you perform the scan. Once the scan policy is added you can modify or remove them. Also, you can directly import or export the scan policy.

Updates and Support: Nemasis provides the application update both automatically and manually. There are regular updates on the new vulnerabilities that are being added to Nemasis database. Along with online update, Nemasis also provides offline updates for the air gap systems. We offer 24x7x365 free online technical support to customers through email, phone, and live chat.

Critical Vulnerabilities Targeted

SQL Injection: The SQL Injection attack is done without changing anything on the database side. To execute this, the test uses an incomplete (or incorrect) SQL statement which will cause the SQL server to throw an error. A hacker can then modify the payload to perform an actual attack. Since SQL commands are injected in an application form into the database, it is possible to change or dump the database's sensitive data (like credit card details or passwords) so that they will be visible to the hacker. It is used to delete or change data or allow hackers to sidestep a login form without needing to guess the password. The solution for this is using stored procedures or parameterized queries to prevent hackers from altering the queries or prevent injection flaws.

Broken Authentication and Session Management: Various types of vulnerabilities related to credentials authentication, session IDs exposure, and sending them through unencrypted connections will allow hackers to either capture or bypass the authentication that is used by the web application. To avoid these kind of breaches:

- Credentials should be protected using hashing or encryption
- Session IDs should not be exposed in URLs and should timeout,
- Session IDs should be recreated once the login is successful
- Credentials and session IDs should not be sent over unencrypted connections.

While it should also look for the password length and complexity, password and username enumeration, protection against brute force login. Also, the organization should use multi-factor authentication such as FIDO alliance and more to prevent the risk of compromised accounts.

Cross-Site Scripting (XSS): This is a web-based attack executed on the vulnerable web applications, the victim here is the user and not the application, and the malicious content is delivered to the user through JavaScript. To avoid this kind of attack web application should filter the user input to discard characters such as < and >. Also, you have to make sure that your server does not display error messages containing input received from the user.

Broken Access-Control: This is mostly poorly configured or missing restraints on authenticated users which enable them to access unauthorized functionality or data, such as viewing sensitive documents, accessing other users' accounts, and modifying data and access rights. Nemasis-DAST detects where access controls are missing and the solutions to overcome them.

Security Misconfiguration: Security Misconfiguration attacks exploit configuration defect found in web and application servers. Administrative or debugging functions may be accessible to anonymous users which may provide a means for a hacker to bypass authentication methods and gain access to sensitive information which may require special privileges. Misconfigured SSL certificates and encryption settings, the use of default certificates, and improper authentication implementation with external systems to gain unauthorized access or knowledge of the system.

Sensitive Data Exposure: The most common defect is simply not encrypting sensitive data. When crypto is used, weak key generation and management, and weak algorithm usage are normal, especially weak password hashing methods. Browser weaknesses are very common and easy to identify, but difficult to exploit on a large scale. Hackers have difficulty in identifying server-side defect due to limited access and they are also usually hard to exploit.

Insufficient Attack Protection: Most applications and APIs identify invalid input, yet essentially dismiss it, giving the hacker chance to attack again and again. Such attacks indicate a malicious or compromised user probing or exploiting vulnerabilities. Identifying and blocking both manual and automated attacks is one of the best ways to increase security. Hackers use various automated tool and skilled humans to identify vulnerabilities and possibly exploit them.

Cross-Site Request Forgery (CSRF): CSRF is an attack that includes forcing a victim to send an HTTP request to a target destination without their insight or intent in order to perform an activity as the victim. The idea of the attack is that CSRF exploits the trust that a website has for a user. CSRF attacks are not necessarily cross-site, yet they can be. CSRF is possible due to various factors such as the victim has an active session, is authenticated via HTTP Auth, or on the same local network on the target site. The risk of data disclosure is drastically increased when the target site is vulnerable to XSS because XSS can be used as a platform for CSRF, allowing the attack to operate inside the limit of the same-origin policy.

Using components with known vulnerabilities: The popularity of this issue is far-reaching. Some known vulnerabilities lead to only minor impacts, some of the largest breaches to date have relied on exploiting known vulnerabilities in components. Depending on the assets you are protecting, perhaps the risk should be at the highest priority on the rundown.

Under-protected APIs: Nowadays, modern web applications and APIs are increasingly made of rich clients (browser, mobile, desktop, and more) that

connect to backend APIs (XML, JSON, RPC, GWT, custom). APIs (microservices, services, endpoints) can be vulnerable to the full scope of threats. Unfortunately, dynamic and sometimes even static tools don't work well on APIs, and they can be difficult to analyze manually, so these vulnerabilities are often unfamiliar.

Reports

- **Scanning report-based on the Threat Level (High, Medium, Low or Information-only) of all the affected URIs:** The report will contain the list of affected URIs and the threat to which it is exposed to. It is listed on the basis of threat-level, that is, high, medium, and low. Vulnerabilities with high threat level are categorized as the most dangerous, which put the target scanned at maximum risk for hacking and data theft. While vulnerabilities with medium threat-level are caused by server misconfiguration and site-coding flaws, which facilitate server disruption and intrusion and vulnerabilities with low threat-level are derived from lack of encryption of data traffic or directory path disclosures.

- **Remediation for discovered vulnerabilities:** This provides a list of vulnerabilities identified while performing a scan on the web applications. It also prioritizes the risk of the vulnerability detected and provides the remediation plan for them.

- **Threat Severity Classification (WASC):** This report displays WASC threat classification issues found on your web applications or website. The Web Application Security Consortium (WASC) is an international group who produces open-source and widely agreed upon best-practice security standards for the Worldwide Web. If any vulnerability is identified, performing the attack requires using at least one of several application attack methods. These methods are generally referred to as the class of attack. The WASC web application threat severity classification has a list of attacks which includes Brute Force attack, XXS, DoS, Path Traversal, and many more.

- **Weakness Classification (CWE):** The Common Weakness Enumeration Specification (CWE) is a community-developed list of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts. The CWE has the list of common software weaknesses which serves as a process for describing code vulnerability assessment capabilities in terms of their coverage of the different CWEs.

Note: Nemasis DAST license model is provided as a subscription for a single domain.