Prevent Data Leaks Caused by Web Applications – eScan

Security Researcher Elliot Alderson has discovered a huge leak of Aadhaar numbers from Indane’s website as well as app. The leak has not just put Aadhaar number of 6.7 million people at stake but also their personal details.

Data breaches like these do not necessarily translate into a complicated attack by hackers, they may be simple attacks attributed to configuration errors or unpatched systems or coding error. However, when the same issues are discovered by Security Researchers, who put in efforts to find a way into the system and following the processes laid down for responsible disclosure, then these are termed as bugs / vulnerabilities.

Organizations lately have been hiring researchers to find vulnerabilities in their networks / systems with the objective to ensure that their systems are protected and decrease the footprint of the attack surface that may otherwise allow easy access to the hackers.

Existence of vulnerabilities in Web Applications, pose a greater risk to the integrity of the entire system, while mobility being the new mantra, we are embracing web-application and are moving towards cloud based systems at a much greater pace than ever before. Web-Applications viz, ERP, CRM, Emails are a conduit to the data which is essential for the functioning of an organization and unlike their stand-alone / networked counterparts, have a much larger attack surface.

Search engine caching paradigm

According to eScan, there have been instances when confidential datasets have been cached by Google, which otherwise shouldn’t have been left exposed for general public to view and for hackers to gain foothold into the system.
Caching of datasets by search engine is an excellent example to the non-existence / therein lack of Authentication and Access Management, which is an inherent requirement when data is being served. The recent leak which was exposed by the French Researcher “Elliot Alderson” was cached by Google’s Search Engine, furthermore, this cached dataset was devoid of authentication would have allowed anyone with scripting knowledge to automate the data-mining tasks on the actual urls. Presently, as per our observation, the erring URIs has been taken down by Indane.

**Could this have been averted?**

Dynamic Application Security Testing facilitates organizations to test their web-based applications / websites for existence of vulnerabilities viz. XSS, CSRF, SQLi, LFI / RFI, furthermore, pages which are supposed to be accessible after authentication are also tested for vulnerabilities which may allow users to gain access to the other resources served by the application.

In this case Unauthenticated API Endpoints were solely responsible for allowing anyone with the knowledge of using Google Search Engine to gain access to the datasets, which otherwise should have been protected.

DAST Solutions also facilitate organizations to discover such vulnerable end-points not just by crawling the web-application by also by querying and analyzing the results of the search engines.

The need of the hour for all organizations which handle sensitive data should conduct regular DAST audits of their web-applications.