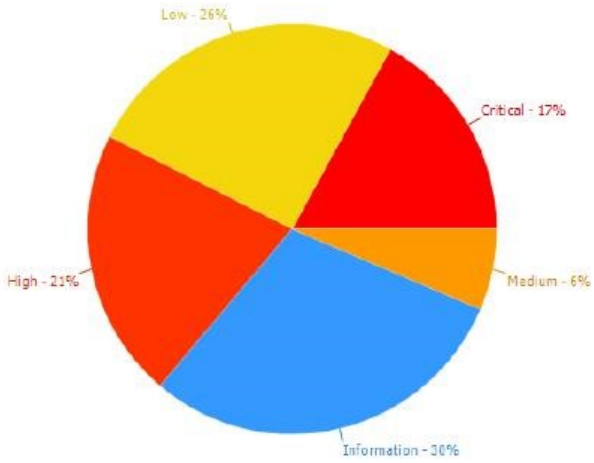
PROTECTING WEBSITES FROM VARIOUS ATTACKS

Qodirov Farrux Ergash o'g'li student
farruxbek0209@mail.ru
Mansurova Zarina Anvar qizi student
Karshi branch of the Tashkent University of Information Technologies named after Muhammad al-Khorezmi

Web vulnerabilities today outnumber any other information security problems in terms of number and associated risks. 10% of web applications 10 years ago - this is almost 80% of the total number of corporate applications today. The web is not only on the Internet, ERP, ABS, network management and much more use web technologies. The overwhelming majority of external attacks on corporate information systems are aimed precisely at the vulnerabilities of web applications, and with a multiple increase in risks, special attention has been paid to identifying and eliminating vulnerabilities in them. And if the web is accessible to the attacker within the network, its tasks are simplified by an order of magnitude.



Vulnerability	Suggested Action
Out of Band Command Injection	**Fix immediately:** With these vulnerabilities your website could be hacked right now. You should make it your highest priority to fix them immediately. Once you've done this, you should rescan to make sure you've eliminated them.
Remote File Inclusion	
Code Evaluation (PHP)	
Blind SQL Injection	
Boolean Based SQL Injection	
Out of Band Code Evaluation (PHP)	
Blind Command Injection	

Pic1. Web application vulnerabilities

Security Gateway Web application security is different from network security, even if you do not take into account the single point of failure of UTM and its possible impact on response time and network bandwidth, and, consequently, reduced availability. Web applications should be accessible to all, so

it remains only to allow all incoming traffic to ports 80 (HTTP) and 443 (HTTPs) and hope that everyone will play by the rules. Monitoring sessions for the presence, identification and blocking of executable code does not replace the analysis of web application traffic, therefore, exploiting the vulnerability through a legitimate web request is easy if you have an all-in-one security gateway. Myth # 2: Web Application Firewall (WAF) Web Application Firewall (WAF) analyzes HTTP / HTTPs web traffic for intrusion detection. For example, if an attacker attempts to exploit a known web application vulnerability, WAF can block the connection, but there are nuances: The WAF does not solve the security problems of the web application, it will only stop part of the attacker's requests to the application. As good as a WAF administrator, the tool itself, which is a highly configurable software / appliance, is just as effective. Again the weakest link in the chain is the user. The firewall will not be able to help if it is not configured / trained professionally and permanently. WAF is a common software with its own vulnerabilities and bugs. Security experts regularly identify all new WAF vulnerabilities that allow access to the administration console, disable or bypass the firewall, which represents an additional layer of protection, but do not provide a solution to the problem. It makes sense to apply WAF after evaluating the security of web applications. Myth # 3: Web application security network scanner Network security scanners are designed to detect unsafe configurations, the lack of necessary updates and server and network device vulnerabilities, and not web application vulnerabilities. The architecture of solutions, a huge number of rules and signs to be checked when scanning a network still sometimes allow manufacturers of network scanners to offer additional functionality for finding vulnerabilities in web applications under a separate license (for example, from Qualys, Tenable, etc.) or even for free. But the usability and quality of their work are far from even the average level of professional web application scanners, and confidence in such products cannot be restored after finding critical vulnerabilities where the universal scanner has worked 100%. Fuzzing tools, protocol testing, exploit search, attack generators, memory analyzers and debuggers are also not complete tools for analyzing web application security.

How to protect web applications? To ensure that a web application is secure, you must identify all the security problems and vulnerabilities in the web application itself before the attacker identifies and uses them. It is very important to regularly perform the process of detecting vulnerabilities in a web application throughout the software development life cycle (SDLC), and not only during the operation. Testing in the early stages of development is of paramount importance, since it may be very difficult or impossible to ensure the security of the application in the future without rewriting it. The earlier web application security is included in the project (by design), the more secure the web application will be and the cheaper and easier it will be to fix the identified problems at a later stage. There are several technologies for detecting vulnerabilities in web applications: automatic scanning on the white box principle (white box); manual verification of the source code; penetration test; automatic scanning on the black box principle. The best of

them does not exist - each has its own pros and cons. According to Gartner (Magic Quadrant for Application Security Testing 2018 dated March 19, 2018), the following are distinguished: static testing (SAST, Static Application Security Testing); dynamic testing (DAST); interactive testing (IAST); mobile application testing (MAST). Moreover, Gartner authoritatively does not distinguish between technology, comparing non-related in terms of functionality and purpose of the solution, but it seems, comparing the volumes of client bases of manufacturers. For example, it gives the leading position to HP and excludes from the review PortSwigger (who does not know Burp), draws an analogy between Qualys, the well-known Web Application Scanning (WAS) network scanner, with the Rapid7 AppSpider web vulnerability scanner software. It's nice that the domestic Positive Technologies Application Inspector got into the review. Only in their description honestly made a distinction between SAST and DAST. The positioning by some manufacturers of their solutions as interactive scanners (IAST), as a rule, implemented as agents of software test environment, should not be misleading. IAST supposedly combines the best features of SAST and DAST, but these are products of various classes in essence ("… as a stand-alone product" (C)). Also, separate mobile application scanners (MAST) should not be allocated as almost all modern SAST and DAST are suitable for scanning mobile applications. Nevertheless, we will try to adhere to the proposed terminology. Static source code analyzers (SAST, white box scanner) check source (and / or binary) application codes for detecting intentional (NDV) and unintended errors in software. They are successfully used by developers who have access to the entire code, but complicate development procedures. The scanner will detect technical vulnerabilities, but does not identify logical vulnerabilities that can only be identified through manual auditing. These types of vulnerabilities cannot be identified using an automatic tool that does not have intelligence. Automatic scanning should always be accompanied by a manual audit. On the other hand, the manual process takes a considerable amount of time (weeks and months) and can cost a fortune.

Grant the lowest possible privileges to each application, service, and user. Separate web environment, development and testing environments. The included debugging of the web application environment generates logs containing confidential information about database configuration. Do not upload log files or source code files to the "live" web application environment. Unlinked information (customer account numbers and website user activity) is stored in different databases with different users. Apply the same concept of segregation to OS and web application files. Ideally, the web application files, i.e., the directory that is published on the web server, should be located on a separate disk from the OS. Always use the latest software version and install security patches. Conduct regular monitoring and audit of servers and logs, analyze server log files. In addition to the web application security scanner, be sure to use a third-party network security scanner.

References:

1. https://www.anti-malware.ru/analytics/Technology_Analysis/web-security-myths-and-reality
2. https://www.specialist.ru/course/bezsite
3. http://www.ready.by/publikatsii/soprovozhdenie/bezopasnost-veb-saytov-i-ee-obespechenie.html