Khai Phan Thanh (100901164)

BTN710 – Information Security

1. In class we talked about 4 different types of firewalls. However, those are not the only types. Search the Internet for "Next Generation Firewalls" (NGFWs), and in 250 to 400 words explain how they work, their advantages, disadvantages. Cite your references properly. References are not included in the word count.

* According to *Fortinet,* NGFWs will enhanced the current features of SPI, DPI, VPN, network monitoring and IP mapping to possess a deeper and better inspection method that allows them to quickly identify any threats or possible attack or incoming malware. For corporation, NGFWs will provide more control on the application and IPS. Fortigate, developed by Fortinet will bring the power of IPS, web filtering, SSL and automatic threat detection into every host device. To avoid the expanding of threat in Network Security, Fortigate implemented AL into its products to inspect from unencrypted text to encrypted text (with the industry standard of TLS 1.3) to read and avoid any packet that could contain malware to the host. Fortigate will also provide unparallel traffic inspection speed for enterprise as they usually process lots of data every second. As traditional firewall is limited on the speed and packet handling, this would give enterprise a better option and less likely to cause any delay from using it. With the support of unparallel traffic inspection, Fortigate can help enterprise feel more comfortable when using the NGFWs because it can detects anything from a small malware, up to a ransonware or even any possible attack in just matter of ms.
* In extend from *digitalguardian* website, the main difference between traditional FWs and NGFWs is the ability of analysis and using whitelists signature to identify an authenticate data. Future NGFWs has the ability to working with any existing antivirus program or any other security application (instead of 1 for traditional security), which can add multiple layer of protection.

1. The slides show how an attacker can use trackers to compute the result of queries that correspond to smaller than k records, assuming that the system does not return any results with fewer than k or more than n-k records involved. How can an attacker compute the result of a query that corresponds to more than n-k records?

* A

1. Visit  <https://owasp.org/www-project-top-ten/> and read the top 10 web application security risks. Choose 5, and in 100-150 words explain the risk and a countermeasure that can defend against that type of vulnerability.

**NOTE:** You can write for up to 10 security risks. The extra ones (after the fifth one) can earn you bonus marks. The mark will be considered in question 4, but you need to provide the answer in question 3.

1. **SQL Injection: User can exploit the security by apply SQL query into one of the field which connect to Database, let’s say login function. Solution is that, avoid concatenating variable with predefined SQL query to avoid user sending a query into that concatenated variable. Other potential error is to have SQL query display in the URL, where user could edit and modify to receive a new value back. Solution is to do the DB query secretly in the back-end, front-end display should display a normal URL only, which when the user modify will not affect to the back-end result.**
2. **Broken Authentication: Dependent on password, timeout session doesn’t set correctly and random password guessing are the common issue of this. Solution could be to set a limit timeout session if the website doesn’t detect any interaction from user, apply 2-factor authentication for higher chance of authenticate the correct owner and set a limit on entering wrong password.**
3. **Sensitive Data Exposure: Flaws when using auto encrypting and decrypting function in DB, TLS certificate is out-of-date or no certificate was applied allow the site to monitoring user behaviour and intercept in the middle connection, unsalted or simple hash value applied on password storage. #1 solution is to apply TLS certificate as soon as possible as it can avoid any interception from unknown site interrupt, only decrypting the data when user is truly authenticated (after 2-factor authentication), verify the request and use the decrytion method on HTML page, not in the back-end server. For password, use salted and apply multi layer of hash to avoid decrypting the password.**
4. **Broken Access Control: On person with assigned permission can access certain page and must be authenticated before accessing. If a user is trying to bypass the access by using the URL that an assigned person will get after login to access the data, it should not work as no authentication was detected. Solution is to grouped all of the page required permission and have it covered with an authentication layer. What that means is, authentication step is required on every page when somebody is access, and once a certain person is granted the access, it should place a cookie or retrieve host IP to confirm the authentication if that person redirect to another page also require authentication. When fetching data input from user, ensure to verify the data before input to any of the SQL query, such as input required an integer type but received a string. Also, ensure account number stored in DB must have a unique format and user cannot guess out, and when user tried to input something that does not follow the format, reject the request immediately.**
5. **Security Misconfiguration: Directory listing is still enabled, display meaningless error message (usually display what went wrong in the back-end) to user, doesn’t prompt user to change the permission of data when upload something to a public place are the common misconfiguration security developer usually forget. Directory listing can enable attack to work around the URL to move the location to another place (by using /../secret or similar). To avoid this, only enable this permission to person who is authorized and have been authenticated or best of all is to have it only accessible by developer when they accessing server only. Any attempt to access outside of the allowed zone will be denied. Meaningless error message might be harmless to some user. However, technical user and hacker sometimes can understand the error and figure out a way to work around with that if they know what is the error is pointing to.**
6. **Cross-Site Scripting (XSS): One user can embedded an HTML code to an input box and have it store in the server, when another user get the information from that user, the entire code will be execute on that user’s computer. This is usually happening when security developer doesn’t cross check the input allowed for the box. Since HTML tagging is known as a string, not a coding method, it is the matter of which tagging they use because tagging <script> will allow user to send a script and bypass the checking. Solution for this would be to enable code detection in the input and avoid it, the best option would be to unable special character such as <> or = or any special character that is not required and not usually use by user.**
7. **Using Components with Known Vulnerability: Using a free version of API or components in the application sometimes can lead to a serious security issue that nobody is aware. Such API or components when implemented might required the highest permission from the application and most of the time, the app will just give the permission without the developer notice. Once the permission is granted, they can secretly retrieve any input information and send it back-door to the owner. These flaws can be intentionally or accidentally depends on the sources of it. This vulnerability requires somebody who have a lot of experience to dig into the code to find any flaws on that. However, sometimes removing the flaws can cause the component to stop working too. At the point, there isn’t a clear solution on how to fix this, as some big company already develop their own components to avoid that issue, but developing will be costly for small company.**
8. **Insufficient Logging & Monitoring: Missing logging activity is a dangerous approach for any security developer. Logging activity will allow developer to monitor what is happen in the server when there is a connection and request. Security developer should monitor the logging and identify and suspicious activity, such as attempt to login into an account multiple times, an attempt to gain access and control the server, or any action that interfere with the server. Wireshark is one of the application that allow any user to monitor the connection happen between the user and a host via a real-time logging. Solution is to implement the logging system and have it run at all time. Use keyword or any sorting method to detect anything suspicious or anything could potentially become an attack.**

**References:**

* <https://www.fortinet.com/products/next-generation-firewall>
* <https://digitalguardian.com/blog/what-next-generation-firewall-learn-about-differences-between-ngfw-and-traditional-firewalls>
* <https://www.hacksplaining.com/prevention/sql-injection>