Chritopher LaJon Morgan
Nov 16, 2013

Homework 12 CS 465

Question 1	**Question 2, Small Definitions all from (owasp.org)[2]**
Cache Poisoning	Maliciously constructed response that get inserted or stuck into a web cache. Magnified damage until the cache times out. This is a rather hard attack to perform in practice.
LDAP Injection	Lightweight Directory Access Protocol. Occurs when user input is not properly sanitized and used to determine LDAP quires. Using a local proxy it then becomes possible to execute arbitrary commands or grant permission, etc.
Regular expression DoS – ReDos	Some regular expression detection algorithms can run exponentially with respect to their input. This attack utilizes this info to cause the algorithm to get in these very expensive loops and thus hang for long periods of time.
Path Manipulation	When user input is user to specific path information for an operation could allow an attacker to overwrite important system files.
Cross-User Defacement	An attacker can make a request that will cause the server to generate two responses. The second can be erroneously interpreted as a response to a different request.
Session Fixation	An attacker tries to force a known session ID on a user, once the user authenticates the attacker has a valid session. This occurs when the servers doesn't invalidate old session id before authenticating a user.
Deletion of data-structure sentinel	Removal or accidental removal of a data structure sentinel, such as a null terminating string.
Truncation error	e.g., casting a double to an int
Trust Boundary Violation	Occurs when the boundary between trusted data and none trusted data is mottled. Usually happens when both types of data are stored in a single data structure.
Capture-replay	Occurs when a user is able to capture network traffic by simply listening in and then replay part of it back to the server with the original effect.

Question 2: I took time to read each of the above articles and their examples and then tried to fill out this table with my own words, that took an hour. Extending my study of LDAP Injection, I also learned that this attack is very similar to SQL Injection. The idea being that LDAP is just a protocol around a database. One simple example for an attack: When asked for a user name just provide '*'. Depending on the query formation, this would cause the server to return all of the user names in the database. Therefore, the idea is that, unless the user input is cleaned, an attacker can run arbitrary LDAP commands and potentially see anything in the database that he wants to. The attacker could also make unauthorized modifications to the data or cause other unintended behavior.

One suggested way of dealing with this type of attack is to specifically white list good inputs or to only "accept known good."[1] The idea here being to strip out anything that shouldn't be, for example, if we expect the data to be a name one could restrict the size of the input, alpha characters only, no special characters, etc.[1] The reason some suggest white listing over articulating or identifying bad instances is because the search space for correct instances is much smaller than that of bad instances. If one is looking for bad cases, it is likely that one will slip past. … **[1]: http://cwe.mitre.org/data/definitions/90.html [2]: https://www.owasp.org**