Session Attacks

Сесиите предоставят възможност на Web приложенията да съхраняват информация на сървъра вместо винаги да я изпращат до потребителя, така те стават като контейнери за информация. В сесиите много често се запазва потребителското име на клиента а понякога дори и неговата паролата, като по този начин приложението намалява предаването на важна информация и в същият миг създава доста по-приветлива среда за себе си. Поради важността на информацията пазеща се в тези сесии, те са много привлекателна цел за кракери.

Какво дефакто са сесиите? Сесиите това са файлове създадени на локалният сървър които се различават един от друг по session id-то си. Това session id се предава на клиента. Поради технически причини в повечето web сървъри е невъзможно да се установи дали даден клиент е имал започната сесия без той сам да я посочи. Самото посочване става когато клиента представи пред сървъра session id на сесията която той е бил започнал. За да се получи този ефект Web приложенията използват 3 начина за връщане обратно на това session id. Първият начин е session id-то да се вкара като променлива в URL-а и така да се връща обратно към сървърът, вторият начин е session id-то да бъде вкарано като hidden form fields и така да се връща обратно а третият начин това е да се използват Coockies.

Предимства и недостатъци на трите начина

Първият начин е най-лесен за използване, но за своя сметка веднага показва на потребителят, че той използва някакъв вид сесии и поради това се смята за най-несигурен.

При hidden полетата в интегрираната форма (зависимост от вида на изпращане) пак има вероятност клиентът да види това, че се ползват сесии.

А при coockies съществува проблемът клиентът да не поддържа или да е забранил този вид променливи в browser-а си, но от друга страна ако се приеме, че те са разрешени този начин на предаване на session id-то се смята за най-сигурният от трите.

Както споменах по-горе сесиите се разграничават една от друга по своя session id, този session id представлява комбинация от символи подбрани от сървърът. Ако тази комбинация не е много сложна то тя лесно може да бъде налучкана от потенциален нарушител. Този вид атаки се нарича Bruteforce. Втори вид атака срещу сесиите това е session hijacking. При този вид атаки нарушителят успява да се сдобие с легитимен session id като го взима чрез някакъв скрипт от coockies на клиента, подслушва му връзката или по някакъв начин успява да се добере до мястото където се съхраняват сесиите на сървъра(това е страшно опасно ако пазите важна информация в сесийните файлове). И третият вид атаки това са session fixating. Както името казва това са атаки които целят предварително нарушителят да фиксира session id-то на клиента(без той да се усети) и след това нарушителят директно да ползва този session id. Няколко примера на session fixating атаки:

Приемаме, че на domain.tld имаме Web приложение което работи със сесии. Нарушителя създава сесия на сървърът още преди клиентът да е подал заявка към сървърът(съвсем просто като посещава http://domain.tld/login.jps?sid=1234). След което нарушителят намира начин да

прилъже нищо неподозиращият клиент да влезе с тази сесия на сървърът (примерно чрез web търсачка или fake E-Mail). И щом клиентът веднъж се ауторизира пред Web приложението то и нарушителят ще има същите права като клиента.

Друг тип атака от този вид е да се излъже потребителят да отвори страница на която нарушителят е сложил скрипт който автоматично слага coockie с sid=1234 при което нарушителя ще трябва само да почака след това клиента да влезе на страницата.

Защита от тези атаки

1. Създаване на login форма която не създава сесия преди потребителят да бъде ауторизиран.
2. Заключване на текущата сесия само към IP адресът на клиентът който я е създал
3. Ако се ползва SSL, да се заключи текущата сесия и към SSL сертификатът текущият на потребител.

И тук вече един лично мой трик който ползвах при последната си разработка като добавка. Apache разполага с един модул Unique ID. Енкриптвам в base64 unique id-то на заявката + session id-то в една променлива и ги запазвам в сесията и вместо session id всеки път пращам тази променлива и след decode проверявам дали това е правилната сесия. Така ако кракерът не е съобразителен може и да не се усети, че това дефакто не е session id-то а и самата стойност се променя след всяко натискане на потребителя което допълнително усложнява нещата.

Мариян Маринов
hackman[at]hydra[dot]azilian[dot]net