Beskyttelse

Efter at have fundet sårbarheder i vores system, skal de håndteres dem til en grad der er acceptabel for den skade de kan forudsagde.

Til at beskytte kan fire facer som man gå igennem for at beskytte et system. Disse fire facere er finde sårbarheder, rette sårbarheder, dokumentere sårbarheder og moniter sårbarheder.

Finde sårbarhed

Finde sårbarheder kan gøres ved hjælp af pen testning hvor at sårbarheder bliver fundet.

Rette sårbarhed

der er forskellige måder at gøre noget ved en sårbarhed at fjerne sårbarheden, minimere sårbarheden, give sårbarheden videre eller acceptere sårbarheden. det skal ofte tænkes over her hvor stor en skade en sårbarhed gør samt hvad for ressourcer man har til at mod virke sårbarheden.

Dokumentere sårbarhed

Som alt andet inden for sikkerhed er det vigtigt at man dokumentiser hvad man gør for at modvirke sårbarheder og hvordan de skal modvirkes så det er muligt at komme tilbage til ens sårbarheder nemt kunne få et overblik over sårbarheden og kunne evaluere sårbarheden fra ens nye standpunkt.

Hvis muligt logge sårbarhed

Det er vigtigt at logge alt former for angreb systemet er under da information omkring angriberen kan give information om hvor stort pres man er under samt hvem der angriber systemet. alt information kan hjælpe med at evaluere fremtide sårbarheder.

General beskyttelse

På grund af at systemet brugte the MySQL database og ikke havde beskyttede sig mod sql injection, som er en af de mest brugte typer af angreb ifølge owasp. For at beskytte mod sql injections lavede vi alle sql kald om til prepared statements. Prepared statements gør det muligt at beskytte mod sql injections ved at stoppe bruger inputeted data fra at interagere med programkoden.

Vi har gjort det muligt at rapporter filer eller kommentere hvis bruger mener at de ikke burde være på siden. Dette skal hjælpe administratoren og moderatiore til at sikker at forkerte kommentare eller filer ikke bliver lagt op på siden.

Blokering af ip ved unormal aktivitet

Lukke porte (portsentry)

System hardening

Nå et system bliver brugt direkte ud af boksen er det ikke sikkert, da der er sårbarheder for en maskine selv hvis den er opdateret når den kommer ud af boksen og ikke har de rigtige indstillinger til hvad maskinen skal bruges til. For at beskytte mod angreb mod der bruger ens standard indstillinger, kan man system hardning. System hardning er at gå gennem systemet og beskytte angreb som bruger bestemte indstillinger på maskinen. Den bedste måde at harden en maskine på er at læse guides på hvordan forskellige grupper mener man skal harden på og være kritisk overfor deres anbefaleres. Der findes også programmer som går igennem ens system og hardener maskinen samt kan hente opdateringer for de nyeste hardning anbefalinger. Derefter skal processen for at harden systemet selvfølgelig dokumenteres hvad bliver gjort og hvorfor. Som en del af hardning kan reporten for et pentesting angreb bruges til at finde ud af hvad for åbninger systemet har derefter er det så at finde måder at harden må disse åbninger hvilke disse åbninger kan blive middigeret eller fjernet fuldt.

For at kunne finde ud af om en angriber er kommet forbi ens forsvar kan man bruge tripwire. Tripwire er et program der skanner en location på computeren for deres nuværende hash værdi. Ved hjælp af hashværdien er det muligt at finde ud af om en fil er blevet ændret. Når man har en overblik over hashværdieren for systemet kan man finde ud af hvad for filer der ænder sig og hvis specifikke filer bliver ændret, kan man finder ud af at man er under angreb. Filer som vi er særlig opmærk som på er password som indeholder information omkring brugeren for systembruger på web-serveren.

Uddyb

Den default bruger for MySQL har adgang til alt i databasen dette inkluderer at lave nye bruger og droppe tables, det er derfor en god ide at minimere skaden der kan ske. Vi vil derfor gerne have bruger som kun har adgang til at gøre hvad de har brug for at kunne gøre deres funktion. Vi har derfor lavet bruger som har minimum adgang til databasen alt efter hvad for en opgave brugeren har, det ville f.eks. kun muligt at tilføje logs til databasen ikke at fjerne dem.

GRANT SELECT ON website.Users TO 'login'@'localhost';

GRANT INSERT ON website.Logs TO 'logging'@'localhost';

GRANT SELECT, INSERT, UPDATE ON website.Comments TO 'comment'@'localhost';

GRANT DELETE ON website.Comments TO 'deletecomment'@'localhost';

Dette minimerer skaden der kan ske hvis hjemmesiden fejler da den ikke har mulighed for at droppe tabeller eller databaser.

Kerberos

Loging ind

Brugeres kodeord var ikke krypteret i databasen, på grund af at kodeord stå i plain tekst gør at det er muligt for alle der har adgang til databasen at kunne skaffe sig brugeres kodeord. Det er muligt for selv ansatte som har adgang til databasen at kunne skaffe sig bruger adgang eller sågar admin adgang. Kodeord bliver krypteret ved brug af password\_hash med argon2 som key derivation function (kdf), et salt bliver også brugt til kodeordet for at stoppe brug af rainbow tables, salten bliver gemt ved siden af kodeordet i plain tekst.

Filsystem

Det har været muligt at over skrive filer da systemet ikke tog højde for at du uploade en fil som hedder det samme som en foregående fil. Dette ladte bruger over skrive filer som var uploadet filsystemet. For at stoppe bruger for at over skrive filer tjekker vi først efter filer med samme navn og extention hvis der er en fil stopper vi filen fra at blive overført.

Den funktion som kunne havde været værd at implatientisere ville være en max leve tid på filer der ikke bliver brugt. Ved at havde en max levetid kan man limitere hvor meget plads der bliver brugt på serveren.

Vores nu værende filsystem tager imod en fil lige meget hvordan den ser ud. for at stoppe malisource brug at fil serveren burde der implementeres et filer på hvad for filtyper der kan uploades til filsystemet så det kun er muligt at uploade filer der har et bestemt filformat. Det er muligt at komme uden om dette filter ved at gemme sin fil som en anden type ved at navngive den som filtypen, filer ville derfor blive nød til at blive testet for hvad for type de er ved at kigge på dataen.

Kommentar

det er muligt at lave cross site scripting på vores kommentarer dette betyder at det er muligt for angriber at effekt kode på siden. Cross site scripting betyder at det er muligt at indsprøjte scripts via kommenter på kommentar siden. Måde hvor på at cross site scripting blev forhindret her var ved at ændre ’<’ til ’ &lt;’ som er synlige identisk men ikke køre kommandoen. Den kan være at man gerne vil havde at det er muligt at bruge html tags i dette tilfælde skal man finde en måde at fjerne desinficer script tags samt fjerne muligheden for at henvise til andre hjemmesider da det ikke vides som de er sikker.

~~Adgang til at ændre kommentar hvis man lavede den? Log gamle kommentar og hvad den blev ændret til~~

~~Vi valgte at gøre det muligt for bruger at ændre deres egne kommentere og administrator at ændre alle kommentar dette gør det muligt for administrator at monerare hvad der bliver skrevet på siden.~~