Øving 5 - sikker kode

Native programs

Når det kommer for mye data til en buffer enn det den kan takle så oppstår det ett problem ettersom at dataen som bufferen ikke takler må ha ett sted å gå. Dette resulterer i at dataen blir overfylt inn i en nærliggende oppbevaringsplass, dataen som kommer inn enten overskriver eller korrupterer det som allerede ligger her. En buffer overflow vulnerability er at en kan få tilgang til systemer en ikke burde kunne. Dette blir gjort ved å overfylle minnet med data, dette gjør at en kan overskrive eksekveringsbanen til ett program. Dataen som også blir flyttet kan inneholde kode som har et hensyn til å utføre spesifikke handlinger.

Det er for det meste C og C++ som har problemer med buffer overflow, dette er på grunn av at dem ikke har noen beskyttelse mot buffer overflow. For å ikke støte på disse problemene burde en unngå å bruke standard bibliotekets funksjoner som ikke er sjekket for grenser. En av disse funksjonene er strcpy. Grunnet dette så er det flere målrettede angrep mot disse programeringspråkene. [1]

Web sites

Valgt 3 svakheter fra *OWASP top 10 2017* liste.

A1 – Injection

Injection er en exploit som ofte brukes til å få tilgang til databaser via data felt på nettsider. En injection kan forekomme omtrent alle steder hvor data blir innputtet. I for eksempel ett kommentarfelt er det mulig til å få tilgang til databasen som lagrer alle kommentarene, og om databasen ikke er organisert med tanke på sikkerhet kan man få tak i annen info for eksempel epost adresser, brukernavn og passord hashes.

Måten en SQL injection blir utført på er ved å prøve å få inn sin egen kode i en SQL Query. For eksempel i ett søkefelt kan man avslutte input stringen og linjen med ‘; etter dette kan man gi

sin egen SQL Query som vil bli eksekvert sammen med søke queryen. Her kan man gjøre mye skade på databasen som å slette eller dumpe elementer i den.

Måten du forebygger dette kan være ved å gjøre noen symboler ulovlige å inpute eller sanitere inputen etter den har blitt innputtet. [2]

A3 – Sensitive Data Exposure

Mange nettsider har mye sensitiv data lagret om brukerne sine og om disse dataene ikke er sikkert lagra kan det være veldig skadelig for brukerne. Personnummer og kredittkort er ting som er svært skadelig om det blir tilegnelig på nette. Typiske feil som gjør dette mulig, er for eksempel lagre data i klar tekst, ikke bruke kryptere protokoller som HTTPS og svake/utdaterte krypterings algoritmer.

A7 – Cross-Site Scripting (XSS)

Cross-Site Scripting er når en bruker kan legge inn egnede skript på en side som vil eksekveres når noen går inn på siden. Dette skjer i for eksempel kommentar felt hvor input teksten ikke blir sanitert godt nok så når noen bruker HTML tags blir det lest av nettsiden som faktiske tags. Med dette kan man også legge in JavaScript scripts som kan være skadelige. [3]

Scripts:

Hver fil I Linux system er tildelt tre typer eierskap:

- Bruker: brukeren er som standard eieren av fila han/hun har lagt og kun eieren har rettighet og har tilgang til filene
- Gruppe: filene er tildelt til ei gruppe brukere og hvert medlem av gruppa har tilgang og rettigheter til å endre eller lese fila
- Andre: alle andre brukere har tilgang og rettigheter til fila. Det betyr at alle andre har tillatelse til fila, så når du setter fila som ''Andre'' så gir du tillatelse til andre brukere eller som da også er referert som, resten av verden

UGO eller User, Gruppe & Others er det same som alle de tre type eierskapa nevnt ovenfor

Alle filer og bibliotek i Linux system har tre tillatelser definert til de tre gruppene som er nevnt ovenfor:

- ‘’Read’’ – refererer til brukerens evne til å lese inneholde i fila
- ‘’Write’’ – refererer til brukerens evne til å skrive eller endre fila
- ‘’Execute’’ – refererer til brukenes evne til å kjøre eller sjå inneholde i fila [4]

References

[1] VERACODE, "VERACODE," [Online]. Available: https://www.veracode.com/security/buffer-overflow. [Accessed 14 10 2019].

[2] The MITRE Corporation, "cwe.Mitre.org," 19 9 2019. [Online]. Available: https://cwe.mitre.org/data/definitions/89.html. [Accessed 14 10 2019].

[3] The OWASP Foundation, "owasp.org," The OWASP Foundation, 2017. [Online]. Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf. [Accessed 14 10 2019].

[4] Linux, "Linux," 8 5 2010. [Online]. Available: https://www.linux.com/tutorials/understanding-linux-file-permissions/. [Accessed 14 10 2019].