\section{Applicatie laag}

\subsection{SQL injection}

\subsubsection{Uitvoering en schade}

Bij een SQL-injectie worden er SQl-statements die een slechte validatie hebben gebruikt om een website binnen te dringen. Deze aanval is zo gevaarlijke en destructief dat deze in 2013 volgens \cite{Cisco2013} zelfs op de eerste plaats stond van meest gevaarlijke bedreigingen voor web applicaties. Via een SQL-injectie kan een aanval o.a. de volgende zaken doen:

\begin{itemize}

\item Authenticatie omzeilen om zo in te loggen op de applicatie met administrator bevoegdheden.

\item Gevoelige/geheime informatie in een database bekijken en gebruiken.

\item Malware plaatsen in de database of op de website.

\item Cruciale gegevens verwijderen.

\end{itemize}

Op de Kali-aanvallersmachine wordt er voor deze aanval gebruik gemaakt van sqlmap. Er zijn meerdere tools om een SQL-injectie uit te voeren, maar in dit voorbeeld wordt deze tool gebruikt. Allereerst moet de naam van de database gekend zijn en dit kan makkelijk gevonden worden door "`sqlmap -u www.baele.be --dbs"' in te typen. Nu dat de naam van de database bekend is, moeten de tabellen gekend zijn en dit gebeurd via "`sqlmap -u www.baele.be -D TestDatabase --tables"' waar "`TestDatabase"' de naam van de database voorstelt. Nu is er een hele lijst aan tabellen zichtbaar en kan er een willekeurige tabel geselecteerd worden. In dit geval wordt de tabel "`Gebruikers"' genomen. Hierna wordt er een willekeurige kolom genomen uit deze tabel via "`sqlmap -u www.baele.be -D TestDatabase -T Gebruikers --columns"'. Nu is er de mogelijkheid om de hele tabel te "`dumpen"' of enkele velden in een lokale map. Dit wordt gedaan via "'sqlmap -u www.baele.be -D TestDatabase -T Gebruikers --dump"'. Nu kan er gebrowsed worden naar de locatie die staat beschreven in het terminalvenster om te kijken wat er allemaal opgeslaan is en daar is te zien dat de hele tabel aanwezig is.

\subsubsection{Bescherming en preventie}

Het spreekt voor zich dat deze soort aanval veel schade kan toebrengen. Gelukkig is het ook mogelijk om een webapplicatie te beschermen tegen mysql-injecties. Deze bescherming is codegericht en is dus belangrijk voor de persoon die de webapplicatie en database heeft aangemaakt. Deze persoon moet ervoor zorgen dat de code voldoende getest is en foutvrij is. Volgens \cite{Angus2005} zijn er verschillende manieren om de SQL-code foutvrij te maken. Dit kan gedaan worden door het crypteren van gevoelige data, de database installeren en gebruiken met een account die zo weinig mogelijk bevoegdheden heeft. Enkel de bevoegdheden die nodig zijn zouden op dat account moeten zitten. Het administrator-account gebruiken wordt dan ook afgeraden. \newline

Wat ook belangrijk is, is dat de data gevalideerd wordt, dat er geparameteriseerde queries en "`stored procedures"' worden gebruikt, dat de data opnieuw wordt gevalideerd in de stored procedure en dat de error boodschappen niets weggeven over de interne architectuur van de applicatie of de database. Hier wordt er niet te diep op ingegaan omdat dit niet echt een taak is voor de netwerkbeheerder maar eerder voor de programmeur of applicatiebeheerder.

\section{Transportlaag}

\subsection{Sockstress DDOS-aanval}

\subsubsection{Uitvoering en schade}

Een fysieke machine kan onbruikbaar gemaakt worden door een simpele aanval genaamd "`sockstress"'. Deze aanval heeft de laatste tijd enorm gewonnen aan populariteit in het hackersmilieu en dus ook in de kringen van netwerkbeveiligers. Deze methode wordt gebruikt om servers aan te vallen over het internet door middel van TCP. Deze methode zorgt ervoor dat het lokale geheugen zoveel aanvragen moet behandelen dat deze langzaam maar zeker volloopt zodat de server vastloopt en onbruikbaar wordt. Dit wordt ook wel een DOS (Denial Of Service)-aanval genoemd. \newline

Op de aanvallersmachine, in dit geval de eerder geconfigureerde Kali Linux-machine, worden er twee verschillende "`command lines (cmd)"' geopend. In de eerste cmd wordt er \textit{"`nmap <ipadres slachtoffer>"'} getyped om te kijken welke poorten van het slachtoffer die open zijn. De open poorten worden dan ergens genoteerd want deze zijn later nog nodig. Nadat deze zijn genoteerd, wordt er een script genaamd \textit{"`./arppoi"'} geopend in dit terminalvenster. Dit scriptje is te vinden op het internet en de code is te zien in de appendix. De bedoeling van dit script is ARP spoofing. ARP spoofing is een techniek die door veel hackers wordt gebruikt en waar er vermomde ARP-berichten in een lokaal netwerk worden verzonden. De bedoeling is om het MAC-adres van de aanvaller te associëren met het IP-adres van een host, bijvoorbeeld een default gateway of server, zodat al het verkeer dat bedoelt is voor dat specifieke adres naar de aanvaller wordt verzonden. \newline

Nu dat scriptje draait in het een terminalvenster, hoeft er in het andere venster maar één lijntje ingevuld worden. \textit{"`./sockstress -A -C -1 -d <IP van target> -m -1 -Ms -p <alle opgeschreven poorten> -r 100000 -s 172.16.246.0/25 -vv"'}. Dit werkt ook alleen maar als sockstress is gedownload en als je navigeert naar de sockstress-map. Nu kan er gekeken worden naar de server en is er te zien dat het RAM-geheugen dat in gebruik is op de server exponentieel aan het stijgen is. Als dit de maximume waarde bereikt dan zal de server vastlopen en kan er niets meer op gedaan worden. De engiste manier om de server terug aan de praat te krijgen is door manueel de uit-knop in te drukken en hem dan weer op te starten.

\subsubsection{Bescherming en preventie}

De best practices die op de server zijn geïmplementeerd zijn in dit geval niet voldoende en dus moet er een oplossing gevonden worden. De oplossing in dit geval is vrij simpel. Dit kan gedaan worden door het blokkeren van een IP-adres als het meer dan 10 connecties met een poort maakt in minder dan 30 seconden. Dit wordt gedaan door een simpel lijntje in te typen in de router command line \textit{"`iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 30 --hitcount 10 -j DROP"'}. Aangezien de server enkel handmatig kan worden afgesloten, is de kans reeël dat er gegevensverlies is. Daarvoor is het belangrijk dat dit direct wordt bekeken nadat de server opnieuw is opgestart zodat er direct een restore kan plaatsvinden als dit nodig is. Hiervoor zijn de best practices wat betreft back-ups wel voldoende.

\section{Netwerklaag}

\subsection{Malware applicaties}

\subsubsection{Uitvoering en schade}

Malware is een afkorting van "`malicious software"' en betreft alle software die als bedoeling heeft om een netwerk of computere schade toe te brengen. Er zijn verschillende soorten malware waaronder virussen en spyware behoren tot de bekendste. \citep{Moir2003}. In dit geval wordt er een simpel malware-bestand aangemaakt en op de server geplaatst. Hier wordt er gesimuleerd dat de administrator een schadelijk stukje software download op het internet en deze dan laat uitvoeren. \newline

Door het typen van de volgende tekst in een kladblokbestand kan er een virus aangemaakt worden: "`@echo off :A start virus.bat start notepad.ext goto A"' en sla dit bestand op als "`virus.bat"'. Dit simpel virus zorgt ervoor dat het RAM-geheugen van een computer of server binnen de minuut helemaal volloopt. Het programma start elke keer een commandprompt op en elke keer als dit gebeurd wordt er ook een nieuwe kladblokapplicatie geopend. Dit gebeurd oneindig veel keer tot het RAM-geheugen vol zit en de server of computer vastloopt. Hierna kan een apparaat enkel manueel worden afgesloten om het weer aan de praat te krijgen.

\begin{figure}[h!]

\begin{center}

\includegraphics[scale=0.60]{img/MalwareVoorbeeld}

\end{center}

\caption{Voorbeeld Malware-aanval}

\end{figure}

\subsubsection{Bescherming en preventie}

Het spreekt voor zich dat de best practice voor een anti-virus die eerder al besproken is in 99\% genoeg. Als de anti-virus software up-to-date is en dagelijks wordt geupdate, dan is er zo goed als geen risico dat er bedreigingen op de computer komen. Indien er toch een bedreiging door de anti-virus geraakt, is het aan de persoon in kwestie om verstandig te zijn in wat deze precies open doet en downloadt van het internet. Een veilige extra best practice kan zijn om gebruikers niet de permissie geven om programma's te installeren op de computer. Zo zal er altijd een administrator nodig zijn die zijn toestemming moet geven of een specifiek stukje software mag geïnstalleerd worden.