Een manier waarop een hacker aanvalt

Auteur: drs. M.S.L.F. Manssen
http://www.manssen.eu/

In een doorsnee computer staan programma's opgeslagen op de harde schijf. Een van die programma's is het besturingssysteem, andere zijn b.v. je favoriete office programma (b.v. Microsoft Office of Open Office). Het besturingssysteem zorgt voor communicatie tussen de programma's en de hardware. Verder zorgt het er voor dat er meerdere programma's tegelijk kunnen draaien (multi-tasking) en dat er meerdere gebruikers van de computer gebruik kunnen maken. Om multi-tasking mogelijk te maken, moet het besturingssysteem er voor zorgen dat tegelijk draaiende programma's een eigen plekje hebben in het geheugen, zonder een ander programma toegang tot dat stukje geheugen heeft.

Op het moment dat een programma wordt uitgevoerd, wordt het programma vanaf de harde schijf in het het geheugen geladen. Hoe dat precies in zijn werk gaat, is hier niet van belang. Wat wel van belang is, is dat sommige stukjes geheugen bedoeld zijn om gegevens in op te slaan (b.v. een zin uit een document als het geladen programma een tekstverwerker is), en andere stukken geheugen weer bedoeld zijn om het programma dat wordt uitgevoerd (dus de tekstverwerker zelf) in op te slaan. Deze gebieden zijn strikt gescheiden. Wordt door een virusscanner in een programma een stukje code (vakterm voor uitvoerbare instructie v.e. programma.) aangetroffen dat wijzigingen uitvoert op een stukje geheugen dat bedoeld is voor het opslaan van code, dan slaat de virusscanner gewoonlijk alarm.

Nu kan er een fout in een stukje software zitten, waardoor er per ongeluk gegevens in een stukje geheugen worden geschreven, dat bedoeld is voor code. Stel een programma vraagt om je gebruikersnaam. Het heeft daarvoor 100 karakters in het geheugen gereserveerd, maar controleert niet of de gebruiker meer dan 100 karakters invoert. Elke karakter die de gebruikersnaam langer is dan 100 karakters, kan dan terecht komen in een stukje geheugen dat voor uitvoerbare code bedoeld is. Dit type fout wordt wel een buffer-overflow genoemd.

Als een hacker weet dat er een buffer-overflow in de draaiende software aanwezig is, kan het op de plaats vanaf het 101e karakter i.p.v. een deel van de naam een stukje code plaatsen. Die code wordt dan uitgevoerd onder de naam van degene die de software heeft gestart. (Dit kan de systeembeheerder zijn in het ernstigste geval). De code die de hacker plaatst, is niet triviaal. Ik ga hier niet uitleggen hoe de code wordt bepaald. Het is een serie 1-en en 0-en die een stukje machinetaal voorstellen. Op deze manier kan de hacker software draaien op de te hacken computer, en deze overnemen op het niveau van de gebruiker die het software met de buffer-overflow draait.

Een manier om de buffer-overflow te gebruiken, is via een internetpoort. (Zie mijn artikel over de werking van internet.) De hacker scant dan welke poorten er op een computer open staan. Een bescherming hiertegen is een zgn. firewall. Deze laat niet toe dat er van buitenaf een poort zichtbaar is, wat dus nog meer is dan het niet toelaten van schrijfacties naar een bepaalde poort.

Een andere methode om de buffer-overflow te gebruiken, is door de gebruiker een bestand te sturen waarvan bij het openen een buffer-overflow gebeurt. Daarom is het verstandig niet zomaar bestanden te openen die men b.v. via e-mail binnenkrijgt.