# Scapy

Scapy is een heel machtig interactief programma om op zelfstandige basis een volwaardig internet pakketje aan te maken en deze te versturen. Scapy is een library/module gebouwd op python, wat het zeer eenvoudig van gebruik maakt omdat men geen Type checking doet. Dit maakt het eenvoudiger om met verschillende variabelen te werken, die elks een verschillend doel/type hebben.

Het concept van scapy is zeer eenvoudig uit te leggen voor iedereen die over een basiskennis van het OSI-model beschikt. Het is namelijk mogelijk om op zelfstandige basis een pakketje op te stellen volgens elke laag van dit model, gaande van de “payload” in de Application-layer tot zelf het kiezen van hoe men het signaal verstuurd op fysiek niveau in de Physical-Layer van het OSI-model, elk klein detail kan men manipuleren. Binnen deze lagen kan de programmeur dan ook elke mogelijke eigenschap of parameter aanpassen naar wens. Dit geeft de programmeur dan ook ontzettend veel macht in het internet gebeuren.

Scapy kan gebruikt worden om heel eenvoudige dingen te maken zoals een simpel ping-pakketje, maar voor zij die zich verdiepen kent scapy geen grenzen. Als men weet welke parameters/eigenschappen van bepaalde protocollen moet veranderen in een bepaalde laag van dit OSI-model, is het mogelijk om extreem vieze dingen te gaan doen.

Voor zij die graag weten of zelf bepalen hoe hun “attack” of “tool” zijn werk doet is dit zeer interessant omdat men het heir simpelweg zelf kan instellen. En voor de mensen die graag een cadeautjes achterlaten is het ook mogelijk om hun “tag” in hun pakketjes te steken.

Waar het op neer komt is *“Alles kan! (maar niet alles mag)”* zolang men weet wat men wil en een beetje verstand van de opbouw van een pakketje heeft, kan men doen wat men wil, gaande van ping-pakketjes sturen tot TCP-connecties, ARP-tools, Sniffing scriptjes schrijven, MITM-attacks, en zoveel meer, de mogelijkheden zijn maar zo beperkt als uw kennis.

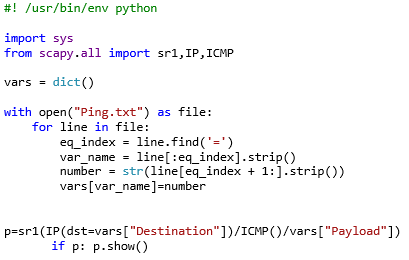
## Scapy in ons project

Wij moesten voor dit project scapy gebruiken om pakketjes op te bouwen op eenvoudige wijzen. Het oorspronkelijke idee is dat men een simpele basislayout in een webpagina toont aan een gebruiker. De gebruiker wordt gevraagd om aantal parameters in te vullen afhankelijk van het pakketje of handeling dat hij wenst te verrichten. Vervolgens als de gebruiker op ‘submit’ duwt zal er een pakketje naar zijn wensen gecreëerd en verstuurd worden. Dit maakt het makkelijker voor mensen men een beperkte kennis over internet-protocollen en pakketjes om toch bepaalde acties te verrichten.

Omdat scapy ook nieuw is voor ons hebben wij slechts één functie in een scriptje kunnen schrijven namelijk een basis ICMP\_ECHO\_REQUEST, dit is het vakjargon voor ‘ping-pakketje’. Wij hebben ook onderzoek gedaan naar andere functionaliteiten, maar die vereiste meer interactie met de gebruiker wat veel moeilijk af te handelen is in een voorgemaakt scriptje.

De opbouw van een ping pakketje in wijn meest eenvoudige vorm bestaat slechts uit 1 essentiële laag nl. de netwerk laag. Binnen deze laag gaat men gebruik maken van het Internet Protocol (IP) en dit encapsuleren in het Intern Control Message Protocol (ICMP). De enige parameters die we in principe nodig hebben voor dit pakketje op te bouwen zijn slechts: het Destination-IP and eventueel indien men wil, een payload met een vriendelijk berichtje.

## Ping script



Eerst beginnen we met de standaard shebang om python module op te roepen, vervolgens importeren we ‘sr1’, ‘IP’ en ‘ICMP’ modules van de Scapy Library. Deze hebben we nodig om het pakketje te kunnen opbouwen en versturen. Vervolgens is er een hele hoop code geschreven die, om de uitleg zo kort mogelijk te houden, ervoor gaat zorgen dat een tekst bestand, waar de parameters, die de gebruiker heeft ingegeven, zijn opgeslagen, uit te lezen.

Nu komt het echte scapy werk. Hier stellen we het pakketje samen. Eerst beginnen we met het Internet Protocol te declareren d.m.v. IP(). Binnen deze ronde haakjes kan men alle parameters van de IP-protocol veranderen, in ons geval enkel het destination ip met de code *‘dst=”ipadress”*’ en omdat wij de variabele destination ip aan de gebruiker hebben gevraagd, lezen we deze uit de tekst file m.b.v. *vars[“Destination”]*. Vervolgens gaan we een laag dieper om binnen het IP-protocol, de ICMP protocol te creëren m.b.v. *ICMP()*, standaard zal de ICMP laag een ICMP\_ECHO\_REQUEST pakketje zijn. Als extra hebben we dan ook voorzien dat de gebruiken een payload kan ingeven als vriendelijk berichte voor de persoon die de ping ontvangt.

Het *‘sr1’* commando zal ervoor zorgen dat het pakketje, dat als parameter wordt meegegeven tussen de ronde haakjes, verstuurd zal worden, en zal er ook voor zorgen dat het antwoord opgevangen wordt en deze in de variabele *‘p’*, zoals hierboven aangetoond, wordt opgeslagen. Vervolgens zal de *‘if-functie’* nagaan of de variabele ‘*p’* wel werkelijk data bevat, m.a.w. of de ping-functie correct afgehandeld is en beantwoord is. Als *‘p’* data bevat zal ‘*p.show()’* ervoor zorgen dat deze data uitgeprint wordt indien men de ping-handeling wil analyseren.

## Conclusie

Wij zelf hebben meer gespeeld met scapy binnen de terminal om het wat te verkennen en hebben veel bijgeleerd. Scapy is zeker een zeer interessante omgeving om mee in te spelen, wij hebben er alvast veel in bijgeleerd.

Als laatste voorstel lijkt het me zeer interessant om scapy te integreren in de labo lessen omtrent OSI-model in de eerste jaren, omdat dit een zeer overzichtelijke omgeving is en men in praktijk kan zien hoe elke laag werkt en wat elke laag van het OSI-model afhandelt en hoe netwerktraffiek werkt. Zo krijgen studenten de kans om een beter inzicht te krijgen over OSI-model, door er gewoonweg wat mee te spelen, ons heeft het zeker geholpen sommige zaken toch beter te begrijpen.

What is scapy?

Scapy is een interactief packet manipulation program. Het is in staat om packets te maken of decoderen met met een ruim aanbod aan protocols. Scapy kan tevens pakketten capture, match request en replies. Nog andere functies van scapy zoals: scanning, tracetouring, probing, unit testing, attacks en network discovery. Scapy voor een python interface in libpcap voor linux of winpcap voor windows. Scapy werkt alleen met python versie 2.x.