I/O VIVAT

JAARGANG 27
NUMMER 2

Computer als Scheidsrechter
Gebruik van techniek waar mensenogen tekort schieten

Node.js
Gebeurtenis gedreven I/O

Ghost in the Wires
Over een meester in social engineering

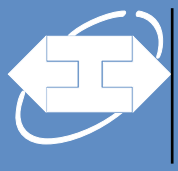
De waarde van "web 2.0"
Een nieuwe dotcom-bubble?

De veiligheid van RSA
Hoe veilig is veilig op het internet?

En verder...

Inter-Actief

Advertentie Quinity



Colofon

Jaargang 27, nummer 2,
januari 2012
ISSN: 1389-0468

I/O Vivat is het populair-wetenschappelijke tijdschrift van I.C.T.S.V. Inter-Actief, de studievereniging voor Technische Informatica, Bedrijfsinformatietechnologie en Telematica van de Universiteit Twente. I/O Vivat verschijnt vier maal per jaar en heeft een oplage van 1800 exemplaren.

Hoofdredactie:
Rick van Galen

Redactie:
Michel Brinkhuis, Ralph Broenink, David Huistra, Ronald Meijer, Herman Slatman, Bas Stottelaar, Niek Tax, Anton Timmermans, Stijn van Winsen

Vormgeving:
Niels Witte

Gastschrijvers:
Marc Hulsebosch, Rom Langerak, Erwin de Moel, Johan Noltes

Voor vragen, suggesties en tips is I/O Vivat bereikbaar via e-mail op vivat@inter-actief.net, twitter op @iovivat, telefonisch op 053-489 3756 of per post:
Studievereniging Inter-Actief
Postbus 217
7500AE Enschede

De studievereniging wil de adverterende bedrijven bedanken voor de samenwerking.

Drukwerk:
Drukkerij van den Bosch & Fikkert

I/O VIVAT

Redactioneel

Is het niet ongelofelijk hoeveel informatiebeveiliging de laatste maanden een enorme opkomst heeft gemaakt als *mainstream* nieuwsonderwerp? Na Stuxnet (en Duqu), hackersgroeperingen en de Diginotar-affaire gaat er geen week meer voorbij zonder dat een securitykwestie de nieuwspagina's haalt. Niet alleen lijken meer van dit soort kwesties op de voorgrond te komen, maar lijkt het 'grote publiek' zich ook steeds meer bewust te worden van hun digitale beveiliging en privacy.

Is dit een terechte zaak? Daar lijkt het wel op. Hoewel berichtgeving in nieuwsmedia niet altijd accuraat is, wordt wel keer op keer aangetoond dat veiligheid niet altijd een goede focus is voor veel ingenieurs. Vergeleken met het afgelopen jaar is digitale beveiliging misschien wel te weinig belicht geweest.

Dat kunnen we in deze Vivat deels rechtzetten. We hebben een artikel over veiligheidscertificaten in navolging van de Diginotar-affaire. We bekijken in hoeverre een webmaster zijn bezoekers uniek kan volgen op basis van verschillende parameters die door een browser worden verschaft – een concept genaamd 'browser fingerprinting'. Verder beschouwen we het boek 'Ghost in the Wires' van wereldberoemde hacker Kevin Mitnick. Als laatste duiken we in de veiligheid van RSA – het algoritme dat wordt gebruikt bij vele beveiligde cryptosystemen.

Een andere trend van de laatste tijd is Node.js, een JavaScript-framework waarmee je snel code kunt draaien op basis van de razendsnelle V8 JavaScript-interpreter. Wat kun je met dit framework, en hoe wordt het in de praktijk gebruikt? Ook twee artikelen daarover in deze I/O Vivat. De variatie in de Vivat wordt tenslotte compleet gemaakt met een artikel over digitale systemen bij sportwedstrijden.

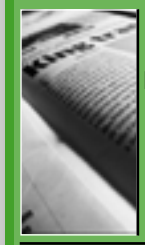
Veel leesplezier,

Rick van Galen
Hoofdredacteur

Artikelen

Columns & ENIAC

Nieuws



Nieuws

Redactie I/O Vivat

BREINCELSIMULATIE, INTEL, CPU, OPT-IN, OPT-OUT, INTERNET, SOPA, SSID-MAPPING

Nieuws

Breincelsimulerende chip gemaakt door wetenschappers

Wetenschappers van het Massachusetts Institute of Technology zijn er in geslaagd een chip te maken die een breincel simuleert. De chip is in staat om te gaan met een belangrijk fenomeen waarbij verbindingen tussen neuronen aangepast worden.

In een normale synaps worden signalen tussen neuronen uitgewisseld door verschillende chemische processen met behulp van ionkanalen en geladen atomen waardoor een puls ontstaat. Hierbij kunnen onderlinge verbindingen tussen neuronen gewijzigd worden.

MIT-wetenschappers hebben de chip zo ontworpen dat de transistors de activiteiten van de ionkanalen kunnen nabootsen. Hoewel de meeste chips met een uit/aan-stand werken, werken deze transistors analoog, net zoals ionkanalen in een synaps. Door de parameters te tweaken zijn ze zo in staat de chip gelijk te maken aan specifieke ionkanalen.

Volgens het MIT is dit een grote stap richting het mogelijk maken van kunstmatig leven. De computer zal steeds meer op de mens kunnen lijken met deze ontdekking. Dit zal echter nog niet op de korte termijn gebeuren, omdat de complexiteit van de hersenen nog steeds niet volledig wordt begrepen. Ook zou de chip gebruikt kunnen worden als hulpmiddel om apparaten met de hersenen te kunnen bedienen. Dat is een concept dat onlangs nog werd getoond door Amerikaanse onderzoekers.

Bron: http://medicalxpress.com/news/2011-10-paralyzed-mind-powered-robot-arm.html

Intel viert jubileum eerste CPU

40 jaar geleden had Intel een merkwaardige primeur: het was de eerste die erin geslaagd was om een ALU (arithmetic logic unit) en een instructiedecoder op een enkel IC (integrated circuit) wist te te integreren. Waar dit in computersystemen voorheen met koperdraden aan elkaar bevestigde afzonderlijke chips gebeurde, was de chip van Intel de eerste die alles in een goedkoop pakket combineerde. Deze oplossing maakte het meteen mogelijk de circuits een stuk sneller te maken dan daarvoor mogelijk was.

Destijds was het opmerkelijk dat Intel met een dergelijke chip op de markt kwam. Intel was eind jaren ’60 bekend geworden als goede geheugenfabrikant, en produceerde sneller geheugen met een goede dichtheid. Het bedrijf was nooit betrokken bij de ontwikkeling van rekeneenheden tot het opdracht kreeg van een Japanse rekenmachinefabrikant om haar chipontwerpen te fabriceren. Daarop realiseerde Intel-ingenieur Ted Hoff zich dat het efficiënter zou zijn om de twaalf benodigde delen op een enkele chip te integreren. Hiermee was de 4004 geboren.

De Intel 4004 was een 4-bit processor bestaande uit 2300 transistors, met een maximale kloksnelheid van 740 kHz. Ter vergelijking is de snelste processor van Intel tegenwoordig is de Intel Core i7 3960X uit de Sandy Bridge E serie. Dit is een 64-bit processor bestaande uit zes cores met een totaal van 2,27 miljard transistors met een maximale kloksnelheid van 3.9 GHz. De Intel 4004 haalde theoretisch 92000 instructies per seconde, terwijl de Core i7 gebenchmarkt is op 177 miljard instructies per seconde Het mag duidelijk zijn dat er veertig jaar overheen zijn gegaan.

‘Amerikaans internet staat op instorten’

In het Amerikaanse congres werd in oktober 2011 een voorstel gedaan voor een nieuwe wet, SOPA genaamd (H.R. 3261). SOPA staat voor Stop Online Piracy Act, en is er, zoals de naam al doet vermoeden, op gericht om piraterij van auteursrechtelijk beschermde materialen op het internet tegen te gaan en strafbaar te maken.

Maatregelen die met SOPA mogelijk worden zijn bijvoorbeeld het door een rechthebbende dwingen van een ISP, om een website die illegaal auteursrechtelijk beschermd materiaal aanbiedt, offline te halen. Hierbij kan gedacht worden aan bijvoorbeeld sites als YouTube, waar materiaal door gebruikers geüpload wordt, waar toevallig audio- of videomateriaal van een rechthebbende wordt gebruikt.

Het feit dat deze wet mogelijk in Amerika aangenomen wordt heeft mogelijk ook verregaande gevolgen voor het globale internet. Als het idee van de Amerikaanse overheid tot goede resultaten leidt zullen er ongetwijfeld overheden in de rest van de wereld zijn die een zelfde wet aannemen. Los daarvan kunnen er ook conflicten optreden wat betreft DNS: sites die in Amerika gehost worden, en die middels DNS onbereikbaar worden gemaakt, zullen ook voor bezoekers van buiten Amerika onbereikbaar worden.

Over de wet is behoorlijk wat ophef ontstaan, natuurlijk van Amerikaanse burgers, maar ook van bedrijven als LinkedIn, Facebook en Google. Zij stellen dat het internet zoals we dat nu kennen zal ophouden te bestaan, mede doordat er geen tussenkomst van een rechter meer nodig zal zijn om een site offline te halen en er dus onbeperkt websites offline gehaald kunnen worden. Dit kan leiden tot een einde van het vrije internet en de mogelijkheid om daar onze mening te kunnen uiten.

Tweede kamer wil Opt-In voor SSID-mapping

Toen in april duidelijk werd dat Google met zijn streetview-auto’s ook gegevens verzamelde van WiFi-accesspoints, was privacy-waakhond CBP ‘not amused’. Google verklaarde dat zij de MAC-adressen niet tot personen kunnen en willen herleiden.

Het CBP liet met een eigen onderzoek zien dat dit wel degelijk kan (waarbij ze per ongeluk de prive-gegevens van enkele Nederlanders publiceerden) en daagde Google voor de rechter. Google kreeg een dwangsom van een miljoen euro en moesten de vergaarde data te verwijderen. Ze stelden daarop een opt-out systeem voor, waarmee het CBP akkoord ging.

Drie maanden later kondigde Google aan dat ze geschikte manier hadden: de toevoeging ‘_nomap’ aan de SSID zorgt er voortaan voor dat de gegevens niet in hun database terecht komen. Daarbij spraken ze de hoop uit dat hun concurrenten zich ook zouden houden aan deze conventie. Echter, in de tweede kamer laaide de privacy-discussie opnieuw op, en zij bleken alsnog niet zo tevreden met een opt-out systeem. D66 noemt het opt-out-plan ‘de omgekeerde wereld’ en CDA dient samen met de PvdA een motie in: volgens hen moet Google vooraf toestemming vragen. De PVV wil zelfs een ‘registreer-me-niet-register’.

Computer als Scheidsrechter

Ronald Meijer
Redacteur I/O Vivat

SPORT, SCHEIDSRECHTERS, CYCLOPS, HAWK EYE, SPORTVISION

Gebruik van techniek waar mensenogen tekort schieten

Een tennisbal die met meer dan tweehonderd kilometer per uur op de lijn stuitert en een lijnrechter die tien meter verderop staat. Het lijkt een wonder dat er dan nog juiste beslissingen worden gemaakt. Atleten kunnen zich geen fouten kunnen permitteren om de wedstrijd te winnen en dit zou bij de arbitrage niet anders moeten zijn. Op amateurniveau komen spelers er onderling wel uit, maar hoe hoger het niveau, hoe meer er op het spel staat.

Men is daarom op zoek gegaan naar een techniek die scheids- en lijnrechters kan helpen minder fouten te maken. Een overzicht van de ontwikkelingen op dit vlak.

Druksensors en stroomdraadjes

De technologische ontwikkelingen begonnen in het tennis, in 1974. Geoffrey Grant en Robert Nicks kwamen met een systeem van druksensoren dat onder het vloeroppervlak gelegd werd. Het systeem kon het verschil detecteren tussen een voetafdruk en een balafdruk, dus samen met een microfoon en een sensor aan het net konden zelfs voetfouten en netservices gedetecteerd worden.

In 1977 werd er een systeem gepresenteerd met geleidende tennisballen en stroomdraadjes op en rond de lijnen. Wanneer er ergens verbinding werd gemaakt tussen twee lijntjes kon bepaald worden of de bal in of uit was. Beide systemen waren erg aan slijtage onderhevig en zijn daarom nooit verder gekomen dan een prototype, maar de toon was gezet.

> "Vliegjes brachten het systeem in verwarring"

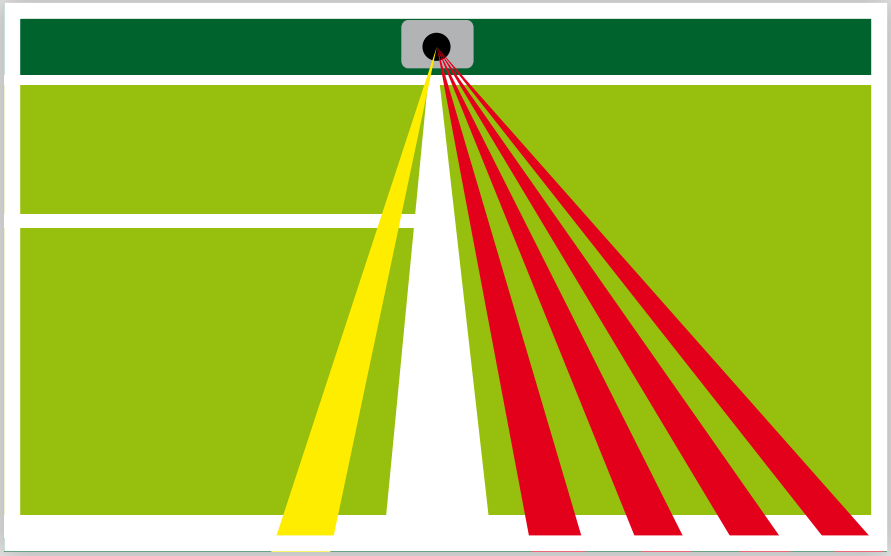
Cyclops en de MacCam

In 1980 werd voor het eerst een systeem op een grandslamtoernooi geïntroduceerd, genaamd 'Cyclops'. Dit waren twee apparaten ter grootte van een flinke schoendoos die in het verlengde van een lijn werden neergezet. Vijf infraroodstralen 'bekeken' het gebied rond de lijn en als de verkeerde straal werd doorbroken, klonk er een harde piep-toon. Cyclops werd vooral in Wimbledon gebruikt en regelmatig verbeterd. Zo duurde het een aantal versies voordat onschuldige vliegjes het systeem niet meer in verwarring brachten.

Pas in 2000 was men de pieptoon zat en werden er hogesnelheidscamera's opgesteld langs de lijnen om, bij twijfel, in een herhaling te kunnen zien waar de bal stuiterde. Dit systeem, de MacCam, werd vernoemd naar de tennisser John McEnroe, berucht om zijn onvrede richting lijnrechters.

Hawk Eye

Vanaf 2007 werden de MacCams op de verschillende toernooien geleidelijk vervangen door het Hawk Eye-systeem. Dit systeem, gemaakt door een Brits bedrijf, gebruikt zes tot tien gekalibreerde camera's die samen de exacte positie van de bal bepalen, op elk moment in de wedstrijd. Het systeem kan zeer nauw-



Figuur 1: De Cyclops met 5 infraroodstralen

keurig het contactoppervlak met de baan berekenen, met een foutmarge van ongeveer 3,6 millimeter (grofweg het dons van de bal). Uitvoerige tests hebben laten zien dat dit systeem in vrijwel alle omstandigheden een juiste beslissing maakt.

Het Hawk Eye-systeem heeft twee tot drie seconden nodig om een beslissing te maken, vandaar dat in 2008 regels zijn opgesteld om misbruik te voorkomen. Elke speler krijgt per set tweemaal de kans om een beslissing van de lijn- of scheidsrechter te controleren met de Hawk Eye. Hoewel spelers in het begin wat moesten wennen wordt het systeem nu door vrijwel iedereen geaccepteerd en bestaat er geen discussie meer over de beslissing.

Scheidsrechters thuis

Het bedrijf SporTVision uit het Amerikaanse Chicago bedacht dat er meer scheidsrechters overtuigd moeten worden dan degene die op het veld staat, namelijk de vele toeschouwers die een wedstrijd via de TV volgen. Dit hebben ze opgelost door grafische info aan het beeld toe te voegen, zogenaamde Augmented Reality (aangevulde werkelijkheid). In Nederland zien we dit wel eens bij zwem- en schaatswedstrijden zoals de naam van de speler in de baan, en soms een lijn waarmee mensen thuis kunnen zien hoe de spelers presteren ten opzichte van het wereldrecord.

In 1996 kwam SporTVision met een systeem dat de puck bij ijshockey traceerde, zodat mensen thuis meer zouden meekrijgen van de snelle bewegingen. Al snel breidde het bedrijf uit met enkele systemen die onder andere tactische lijnen bij American Football op het scherm zetten, of de balbaan bij Honkbal weergeven. Gebruikmakend van eenzelfde systeem als de Hawk Eye kan worden aangegeven of een bal door de

Figuur 2: Extra duidelijkheid bij Honkbal, dankzij sporTVision

Voetbal

Het bedrijf achter de Hawk Eye, Hawk Eye Innovations, heeft al sinds 2007 een systeem klaarliggen om te zien of een voetbal achter de doellijn komt. SporTVision heeft al even lang een systeem om een buitenspel automatisch waar te nemen.

Het lijkt er echter op dat de FIFA voorlopig nog geen technische hulpmiddelen toestaat bij wedstrijden, om de emotie van het spel te behouden. Daarbij wil Blatter, directeur van de FIFA dat het systeem onmiddellijk reageert op situaties, zodat het spel er niet voor hoeft worden stilgelegd.

De directeur van de UEFA gaf in 2007 al aan dat technische hulpmiddelen zoals dit een waardevolle aanvulling op het voetbalspel kunnen zijn, en vanaf 2012 wordt het systeem van Hawk Eye waarschijnlijk in de Engelse Premier League gebruikt.

slagruimte kwam of 'wijd' was, ondanks dat deze ruimte geen fysieke lijnen heeft. Het spel werd zo voor de gemiddelde kijker beter te volgen, maar ook de fanatieke kijker krijgt nu veel meer informatie over het spel.

inmiddels 'changing the game', waarmee ze aangeven hoe belangrijk dergelijke technieken worden voor de sport.

Op dit moment worden de systemen van SporTVision nog niet gebruikt om

Voor de gemiddelde kijker is het spel beter te volgen

Een extra uitdaging voor dit systeem zijn de bewegende camera's. Een klein kastje op de camera verzamelt alle gegevens van de camera zoals de richting, hoogte, zoom, en andere zaken. Samen met de positie van de camera kan berekend worden waar de extra informatie in beeld moet worden gebracht, zodanig dat het soms lijkt of de lijnen op het veld geverfd zijn.

Changing the game

Inmiddels werkt SporTVision met 12 sporten waarbij ze verschillende soorten informatie op het scherm kunnen toveren. Dit heeft zoveel impact op de spelbeleving gehad, dat ze een steeds belangrijkere rol kregen in diverse sporten. De beroemde Nascar-races hebben sinds enkele jaren verplicht gesteld dat alle auto's uitgerust zijn met een gps-tracker, zodat SporTVision de belangrijke auto's uit de massa kan aanwijzen. Sterker nog, de info van alle auto's wordt direct verwerkt tot een 3D-simulatie van de race, zodat fans virtueel kunnen meekijken vanuit hun favoriete auto. Hoewel SporTVision begon met het idee om zo min mogelijk invloed te hebben op het spel zelf, is hun motto scheidsrechters te helpen met hun beslissingen, maar wel om de scheidsrechters achteraf te beoordelen. Zo wordt gekeken hoe vaak ze een andere beslissing namen dan het camerasysteem van SporTVision, dat als referentie wordt gebruikt.

Coaches en scouts

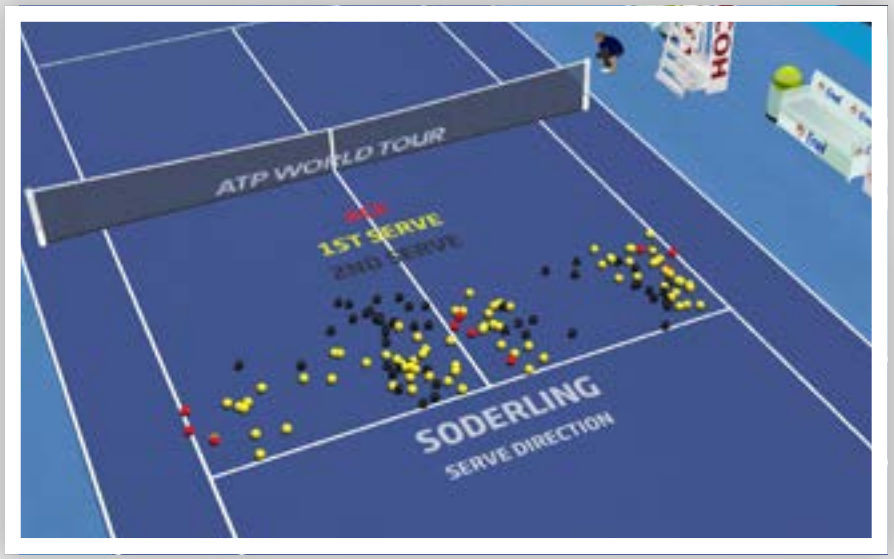
Hoewel de Hawk Eye en SporTVision in beginsel ontwikkeld zijn voor scheidsrechters en publiek, is er een derde partij die veel baat heeft bij alle data die deze systemen kunnen opleveren: de coaches. Met HawkEye is eenvoudig te laten zien waar alle services van de spelers terecht zijn gekomen, of zelfs hoe hoog een gemiddelde topspin stuitert. Deze statistieken kan een coach gebruiken om extra te trainen op bepaalde situaties.

Ook de systemen van SporTVision worden gebruikt door coaches. Zo kopen alle clubs in de Amerikaanse honkbalcompetitie de verkregen data en huren zelfs mensen in om alle data te analyseren. Met de statistieken in de hand kan eenvoudig worden aangegeven op welke worp een pitcher meer moet oefenen, of welk type slag nog niet snel genoeg wordt opgelost door het achterveld.

Recentelijk hebben ook de scouts van de San Francisco Giants de kracht van videoanalyses ontdekt. Het is altijd lastig om een achterspeler te beoordelen die weinig ballen krijgt, maar de Giants hebben een systeem van SporTVision in hun stadion gebouwd dat continu alle spelers analyseert. Bij elke thuiswedstrijd worden dus zowel de eigen spelers als de tegenstanders grondig geanalyseerd. Dit systeem wordt het komende jaar in alle grote stadions geïnstalleerd, zodat alle coaches en scouts van het land over de data van alle wedstrijden kunnen beschikken.

Emotie

Hoewel de techniek hard vooruit gaat en steeds meer kan worden overgelaten aan geautomatiseerde systemen, is er voorlopig nog steeds een scheidsrechter nodig die de eindverantwoordelijkheid heeft. Bovendien zijn er nog veel sporten die zich nog niet laten analyseren door een stel goed afgestelde camera's, en een aantal sporten waar men helemaal geen hulp van camera's wil. Ook voor de kijkers thuis is het maar afwachten of iedereen wel op de harde feiten zit te wachten. Een groot deel van de sportbeleving is de emotie die erbij komt kijken en eigenlijk willen de meeste supporters zèlf roepen dat de Spanjaarden buitenspel stonden in de WK-finale. Toch?



Figuur 3: Statistieken over de service van Söderling

Bronnen

Hawk Eye Innovations
Gemma Voyce
www.hawkeyeinnovations.co.uk

SporTVision
Mike Jakob
www.sportvision.com

DEL Imaging Systems
www.delimaging.com

Advertentie Vanderlande

Ghost in the Wires

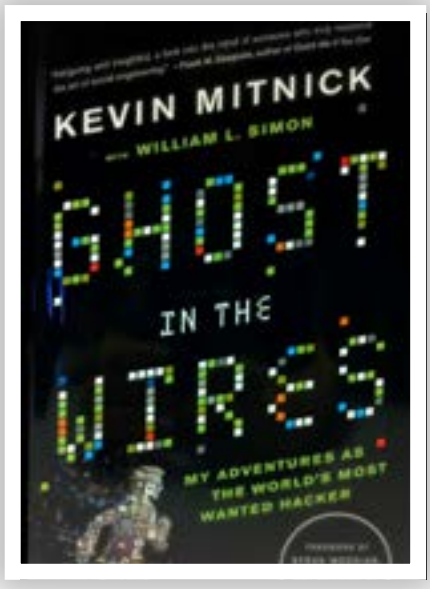

Herman Slatman
Redacteur I/O Vivat

SOCIAL ENGINEERING, PHREAKING, HACKING, COMPUTERNETWERKEN

Over een meester in social engineering

Voor de lezers die aanwezig waren op symposium Immorality zal de naam Kevin Mitnick niet onbekend in de oren klinken. De laatste twee sprekers van de dag noemden hem in hun lezingen en vooral bij de lezing van Jan de Boer over social engineering kwam hij naar voren. Dat is niet verwonderlijk, aangezien Kevin Mitnick in de jaren ’80 en ’90 één van de beruchtste computercriminelen was. Tijdens zijn criminele activiteiten gebruikte hij vaak de kunst van social engineering om ongeautoriseerd toegang te verkrijgen tot diverse computersystemen en –netwerken.

Het verhaal van Mitnick is voor veel mensen een inspiratie geweest. Hele bevolkingsgroepen kwamen in opstand toen hij ten onrechte vast werd gehouden voor misdrijven die hij niet gepleegd had. Een groot aantal scriptkiddies en meer ervaren hackers zagen hem als een voorbeeld. Er werden nieuwe wetten aangenomen in de Verenigde Staten die het inbreken in computersystemen en –netwerken strafbaar maakten. Ook werd er veelvuldig over Mitnick geschreven in artikelen en boeken. Het boek Takedown van John Markoff en Tsutomu Shimomura (deze laatste werd zelf door Mitnick gehackt) werd zelfs verfilmd en in 2000 kon men de criminele activiteiten van Mitnick op bewegend beeld beleven.



Figuur 1: Het boek van Kevin Mitnick

Het verhaal van Mitnick is voor velen een inspiratie geweest

Rondom de film en het bijbehorende boek ontstond een behoorlijke controverse. Enkele van de beschreven hackactiviteiten van Mitnick zouden aangedikt zijn, of zelfs helemaal niet waar zijn. Wellicht was dit één van de redenen dat Mitnick besloot zijn autobiografie, Ghost in the Wires, te schrijven. Het boek, dat als ondertitel ‘My Adventures as the World’s Most Wanted Hacker’ draagt, beschrijft Mitnicks daden vanuit zijn eigen perspectief.

De jonge jaren van een phone phreaker

Kevin groeide op zonder zijn vader en leefde vooral op zichzelf. Op school en daarbuiten maakte hij niet veel vrienden, maar hij was zijn klasgenoten wel ver vooruit op het gebied van wiskunde en spelling. Een buurman van hem was een getraind goochelaar en Kevin werd gegrepen door de magie die hij uitvoerde. Het bedrijven van magie werd voor Kevin de deur naar het misleiden en manipuleren van mensen.

Veel Amerikaanse tienerjongens groeiden in de jaren ’70 op met het zogenaamde phreaking. Dit is een term die gebruikt werd voor het experimenteren of onderzoeken van telecommunicatieapparatuur, zoals de apparatuur in telefoonnetwerken. Men kwam erachter dat de apparatuur op afstand te manipuleren was door bijvoorbeeld speciale nummers te draaien, of door te fluiten op bepaalde toonhoogtes. Ook Mitnick raakte geobsedeerd met het bestuderen van telefoonnetwerken en als snel kon hij gratis langeafstandstelefoontjes plegen of een anoniem terugbelnummer inprogrammeren. Dit laatste gebruikte Mitnick vaak bij het inbreken in bedrijven. Over de jaren werden telefoonnetwerken gedigitaliseerd, en phreaking werd daarmee onlosmakelijk verbonden met hacken.

Telefoons en besturingssystemen

Mitnick maakte een spelletje van het hacken. Nooit was hij uit op financieel gewin noch het moedwillig vandaliseren van zijn doelwitten. Voor hem was

het enige doel om beveiligde informatie te bemachtigen, omdat dat hem een winnaarsgevoel gaf. Toch kwam hij voor het eerst in aanraking met de FBI rond zijn zeventiende voor het inbreken in een systeem dat draaide op tussen twee medewerkers van de National Security Agency. Na deze eerste inbraak op een SCCS duurde het niet lang of Mitnick had toegang tot bijna elke SCCS in de Verenigde Staten.

Mitnicks daden bleven uiteindelijk niet onopgemerkt

RSTS/E, een 16-bits besturingssysteem ontwikkeld door Digital Equipment Corporation. Gewapend met een inbelmodem, computerterminal en een rol thermisch papier verschafte hij zichzelf toegang tot het systeem. Inbreken bleek zeer makkelijk; een account inclusief wachtwoord regelde hij via de zoon van één van de gebruikers van het systeem. Toen de FBI bij hem voor de deur stond kon men hem niks maken omdat er nog geen wetten bestonden die het inbreken in computersystemen strafbaar maakten. Het bezoek van de FBI maakte Mitnick niet bang, en hij bleef inbreken in verschillende systemen.

In het begin waren dit voornamelijk systemen die de telefoonnetwerken ondersteunden. Met behulp van een stukje social engineering; door zich voor te doen als een technicus in het veld, was hij in staat toegang te verkrijgen tot een zogenaamd Switching Control Center System (SCCS). Met een SCCS kon men de hele administratie van een telefoonnetwerk regelen, inclusief het instellen van een trap-and-trace systeem. De toegang tot zo'n SCCS verschafte Mitnick dan ook een krachtig gereedschap en hij wist zelfs een gesprek op te vangen

Continu op zoek naar nieuwe uitdagingen stuitte Mitnick op het door DEC ontwikkelde besturingssysteem VMS. Leergierig als hij was, moest en zou hij toegang krijgen tot de broncode om mogelijke beveiligingslekken te identificeren om zo een achterdeurtje in te bouwen. Een belletje met de ontwikkelingsafdeling van VMS bleek genoeg om toegang te krijgen tot de systemen waarop de broncode en de ontwikkelingstools opgeslagen waren. Mitnick zocht contact met enkele leden van de Chaos Computer Club, omdat ook zij bezig waren met het schrijven van patches voor VMS. Hij kreeg van hen een framework waarop hij verder kon bouwen. Het werd zo voor Mitnick een peulenschil om opnieuw binnen te dringen op elke nieuwe versie van VMS die uitgebracht werd.

Kat en muis

Mitnicks daden bleven uiteindelijk niet onopgemerkt en hij werd door de politie opgespoord in een copyshop in Los Angeles. Hij wist te ontsnappen, maar wist dat hij direct moest verdwijnen om niet binnen zeer korte tijd opgepakt te worden. Hij verhuisde naar Las Vegas

SE Technieken

Social engineering is de kunst van het manipuleren van mensen om hen dingen te laten doen die ze normaal gesproken nooit zouden doen. Het is een techniek die door hackers toegepast wordt om de (vaak) zwakste schakel van beveiliging te omzeilen: de mens. Er zijn enkele aanvalstechnieken die onder social engineering vallen:

Dumpster diving; een techniek waarbij men letterlijk in de afvalbakken van bedrijven rondsnuffelt om allerhande vertrouwelijke informatie in handen te krijgen. Hierbij kan gedacht worden aan roosters voor bewakingsrondes en handleidingen voor computersystemen.

Phishing; men probeert potentiële slachtoffers met een verhaaltje over te halen een bepaalde actie uit te voeren

Baiting; het plaatsen van bijvoorbeeld een USB-stick ergens in een bedrijf. De aanvaller rekent erop dat het slachtoffer nieuwsgierig wordt en de stick in een machine steekt. Hierna zou de machine in zijn geheel over genomen kunnen worden door de aanvaller.

Pretexting; de aanvaller bedenkt van te voren een scenario om gedaan te krijgen wat hij wil. Ook verzamelt hij informatie waarmee hij zich als iemand anders kan voordoen. Het gesprek met het slachtoffer stuurt hij door gebruik te maken van het juiste jargon en zo krijgt hij toegang tot de informatie die hij wil.

en zijn eerst volgende taak was het opbouwen van een nieuwe identiteit. Zijn nieuwe naam werd Eric Weiss, naar de echte naam van zijn jeugdidool, Harry Houdini. Een rijbewijs kreeg hij in handen door zich bij zijn rijinstructeur voor te doen als Amerikaan die net terug was uit het linksrijdende Groot Brittannië. Met zijn nieuwe identiteit op zak verhuisde hij naar Denver, alwaar hij een nieuwe baan vond als systeemadministrateur bij Holme, Roberts and Owen, een internationaal advocatenkantoor.

Ondanks zijn nieuwe baan en stabiele nieuwe identiteit bleef Mitnick, tegen beter weten in, bezig met het hacken van systemen. Zijn volgende grote doelwitten waren niet de minsten. Hij verkreeg de broncode van onder andere SunOS, NetWare, en de besturingssystemen van diverse telefoons van Motorola. Hij was in staat om het nummer van zijn eigen Motorola telefoon te herprogrammeren zodat niet hij, maar de echte eigenaar van het nummer, de kosten van zijn telefoontjes moest betalen. Toen er een nieuwe baas aangesteld werd bij het advocatenkantoor liep het mis. Zij kreeg door dat Mitnick niet alleen voor het kantoor werkte, maar zich ondertussen ook bezighield met andere activiteiten. Mitnick werd ontslagen zonder zijn

woordje te hebben kunnen doen, en hij stond voor de zoveelste keer op straat.

Het was dit moment dat de FBI aangreep om Mitnick aan te pakken. Het werd hem nu lastiger gemaakt om een normaal leven te leiden en hij trok zich verder terug en vertrok naar Seattle. Op een ochtend stond Mitnick, inclusief foto, op de voorpagina van The New York Times. Het artikel getiteld Cyberspace's Most Wanted: Hacker Eludes F.B.I. Pursuit werd geschreven door

Hij had door dat ze zijn GSM signaal peilden

John Markoff, die later bekend zou worden door de verschillende werken die hij schreef met betrekking tot Mitnick. Op een middag, begin oktober 1994, ging Mitnick de straat op. Vanzelfsprekend voerde hij gesprekken via zijn mobiele telefoon, tot een helikopter hem begon te volgen. Al snel had hij door dat ze het signaal van zijn telefoon peilden. De FBI was er op de een of andere manier achter gekomen dat hij telefoons kloonde, en deze gebruikte bij zijn criminele activiteiten. Mogelijk hadden ze hem zelfs al enkele dagen afgeluisterd.

De FBI had hulp gekregen van computerbeveiliger Tsutomu Shimomura, nadat Mitnick bij hem onderzoeksmateriaal voor het reverse engineeren van OKI Telecom telefoons had gestolen. Met hulp van Shimomura kon Mitnick in februari van 1995 aangehouden worden. Een lang verhaal met verschillende rechters en gevangenissen volgde. Gedurende zijn tijd in de gevangenis werd er van diverse kanten hulp geboden. Advocaten wilden het pro bono voor hem opnemen en Eric Corley, alias Emmanuel Goldstein, deed er alles aan om hem vrij te krijgen. Er verscheen een documentaire van Goldsteins hand, Freedom Downtime genaamd, die de hele zaak Mitnick openbaarde aan de wereld. Acht jaar na de eerste keer dat Mitnick werd gearresteerd mocht hij eindelijk weer legaal met een computer werken. Zijn hernieuwde contact met het internet werd zelfs live op televisie uitgezonden.

Conclusie

Gedurende de laatste decennia van de vorige eeuw was het aantal kwaadwillende hackers veel lager dan tegenwoordig, en werd er vooral voor de eer of het plezier gehackt. Tegenwoordig moet de beveiliging van computersystemen goed in orde zijn, want er zijn nu veel meer kwaadwillenden. Dat het niet altijd de computersystemen zijn die falen op het gebied van veiligheid, maar dat er ook een zeer grote menselijke factor meespeelt in informatiebeveiliging is niet nieuw, maar wordt door veel mensen nog over het hoofd gezien. Het was voor Mitnick vaak vrij makkelijk om zich toegang te verschaffen tot systemen door mensen te bespelen en naar zijn hand te zetten.

Tegenwoordig is Mitnick een veelgevraagd spreker op conferenties en heeft hij zijn eigen adviesbureau op het gebied van informatiebeveiliging. Voor diegenen die geïnteresseerd zijn in computerhistorie of computerbeveiliging is Ghost in the Wires een mooie blik op computerbeveiliging in de vorige eeuw.

Bronnen

Ghost in the Wires - My Adventures as the World's Most Wanted Hacker (2011)
Mitnick, K.D, Simon, W.L.

Van de voorzitter

Marc Hulsebosch
Voorzitter Inter-Actief

Marc Hulsebosch zag het daglicht op 15 februari 1991 te Haarlem. Na een succesvolle afronding van het basisonderwijs aan de Albert Schweizerschool te Hoofddorp (waar hij tot zijn studie in Twente gewoond heeft) begon hij aan het VWO op het College Hageveld in Heemstede. Al in de vierde klas ging hij voor het eerst kijken op de Universiteit Twente en nadat het VWO met profiel N&T was afgerond lag de keuze voor de UT ook voor de hand. Hier heeft hij naast zijn studie BIT zitting gehad in verschillende commissies: eerstejaarscommissie, borrelcommissie, symposiumcommissie, businesscoursecommissie, statutencommissie en, buiten Inter-Actief de opleidingscommissie en de Kick-In Delegatie van Stress. Op 11 oktober werd hij voorzitter van Inter-*Actief*.

Symptoombestrijding

Waarde lezers,

Vandaag wil ik het eens met u hebben over het oplossen van problemen. Dat kan op velerlei manieren. Je kunt bijvoorbeeld de symptomen de kop in drukken. “Symptoombestrijding is het bestrijden van klachten, zonder hierbij de oorzaak van de klachten weg te nemen”, zo vindt ons geliefde Wikipedia. Symptoombestrijding is bijvoorbeeld het nemen van een paracetamol na een avondje borrelen in plaats van een verwoede poging te doen om dat adje waterleiding te voltooien. Symptoombestrijding is Rusland dat het probleem van het groeiende aantal zwervers oplost door ze uit de statistieken te schrappen (iets wat Nederland ook schijnt te doen met studenten in statistieken over alcoholisme...). Symptoombestrijding is ook iets waar de UT wel een handje van heeft. De voorbeelden zijn legio.

Wanneer iemand die voor het eerst op de campus komt de snelle rekensom kan maken dat er nooit evenveel fietsen in de standaarden passen als er mensen in de gebouwen zitten, is het logisch dat die fietsen buiten de standaarden terecht komen. De UT vindt dat niks, het staat immers niet zo mooi op je duurbetaalde pleintje. Je kan daarop reageren door een vraag en aanbod dichter bij elkaar te brengen. Je vergroot het aantal parkeerplekken door simpelweg een paar fietsenstallingen bij te bouwen, bijvoorbeeld op de grote leegten van het plein waar de tumbleweeds nog net niet voorbij rollen. En passant kun je dat ook nog gebruiken in je promotie als milieuvriendelijke universiteit als iemand weer komt zeuren dat de UT ’s nachts wel een kermis lijkt.

Een alternatief is dreigen met het verplaatsen van de fietsen totdat mensen ze koste wat kost tussen de bosjes in proppen. Resultaat: een onwerkbare situatie waarin alleen mensen die hun fiets boven hun hoofd uit kunnen tillen op die manier nog uit het doolhof van metaal kunnen ontsnappen. Ben je niet zo sterk, dan heb je pech: om 17:30 wordt je fiets vanzelf weer uitgebouwd door de ijverige studentjes die ook naar huis willen. Als u het vak productiemanagement volgt herkent u het wel: FILO.

Dat zet me aan het denken. Als de universiteit, een instantie die bedoeld is om ons op te leiden, ons zo duidelijk probeert te maken dat het wegnemen van de symptomen voldoende is, moeten we daar dan als studenten niet gewoon lering uit trekken en de wijde wereld ingaan gewapend met deze nieuwe techniek om problemen op te lossen?

Direct zijn veel problemen minder schrijnend. Rijzen de kosten van de langstudeerboetes voor de UT de pan uit? Voor de helft van de boete schrijf ik me in in Delft en lost de UT in één klap hun overmaat aan langstudeerders op. Stond de brug weer eens open en was je te laat op college? Draai je horloge wat bij en zet de tijd naar je hand! Tot slot moeten we natuurlijk niet vergeten hoe symptoombestrijding goed is voor je studie: onvoldoende gehaald? Doorkrassen, een 10 neerpennen en tonen bij de betreffende docent. Op naar cum laude!

Inter-*Actief*

Node.js


Anton Timmermans
Redacteur I/O Vivat

SERVER, JAVASCRIPT, AGILE

Gebeurtenis gedreven I/O

Node.js bestaat al ongeveer 2 jaar en begint steeds meer aandacht te krijgen in de online wereld. Meer bedrijven beginnen te bekijken wat Node.js nou eigenlijk is en er worden zeer geavanceerde applicaties ontwikkeld in Node.js. Wat is Node.js en wat voor interessante applicaties worden er mee gemaakt?

Node.js is gebouwd op Google V8, die precies opkwam op het moment dat de maker van Node.js, Ryan Dahl begon met het maken van Node.js. Node.js is ontstaan vanuit het idee dat gebeurtenis gedreven servers veel makkelijker te maken zijn dan servers die gebruik maken van threads. Ryan wilde graag een server zonder dat de in- en output van de server zorgde voor blocks in de server.

In Node.js maak je voor iedere gebeurtenis die mogelijk optreed, een methode die deze gebeurtenis afhandelt. Node.js zorgt ervoor dat deze methode wordt uitgevoerd als deze gebeurtenis optreed. Dit staat tegenover het thread model, waarin je voor een nieuwe taak een thread aanmaakt die deze taak en mogelijk andere taken uit zal voeren. Het nadeel van threads is de mogelijkheid tot deadlocks en concurrency, daardoor is het ontwikkelen lastig. Om bij threads ervoor te zorgen dat deze problemen niet optreden moet de ontwikkelaar zeer goed nadenken over wat hij doet. Met behulp van Node.js kunnen 'minder goede programmeurs snelle systemen maken.'

De manier waarop Node.js dit doet is met een event loop. Deze werkt constant alle gebeurtenissen door en kijkt of deze zijn opgetreden. Indien een gebeurtenis is opgetreden wordt de methode uitgevoerd die door de programmeur is geregistreerd aan de betreffende gebeurtenis. Als er een tijd niets gebeurt doet Node.js niets totdat het besturingssysteem aangeeft dat er weer iets moet gebeuren. De gebruiker van Node.js merkt hier dus niets van.

> "Matigeprogrammeursmaken snelle systemen"

Er zijn ook een aantal nadelen aan het gebruik van een event loop, de eerste daarvan is dat ook een event loop kan gaan blokkeren als een methode, geregistreerd op een gebeurtenis, zeer lang duurt. De event loop moet namelijk wachten tot de methode klaar is met uitvoeren voordat de volgende gebeurtenis verwerkt kan worden. Het is daarom belangrijk om kleine methodes te gebruiken en als er lang gewacht moet worden hier weer een gebeurtenis voor te gebruiken.

Een ander nadeel van een event loop is dat de code sneller onduidelijk kan worden als er geen structuur in wordt aangebracht. Als er threads gebruikt worden dan heeft de code automatisch een structuur, namelijk methoden die in een andere thread worden uitgevoerd zijn bestemd voor een bepaald doel en kunnen bij elkaar gegroepeerd worden.

Naast het feit dat Node.js gebruik maakt van een event loop is Node.js javascript. Daardoor wordt Node.js zeer toegankelijk voor heel veel ontwikkelaars die al jaren websites ontwikkelen. Javascript in websites is tegenwoordig zeer ingewikkeld en de ervaring van het schrijven van dit soort websites kan weer worden gebruikt in het schrijven van server applicaties in Node.js. Ook kan javascript code die in Node.js wordt gebruikt worden hergebruikt in de website en visa versa. Hierdoor kan de ontwikkeltijd van een webapplicatie behoorlijk worden gereduceerd.

Javascript zelf heeft verder geen mogelijkheden om te communiceren met een database of netwerkverkeer uit te voeren. Deze mogelijkheden worden in een browser omgeving altijd uitgevoerd door een server in een andere taal. Node.js bied deze mogelijkheden als API's aan, bovenop javascript.

Node.js beveelt sterk aan om naast Node.js ook npm te installeren, dit is de Node.js packet manager, dit kun je vergelijken met rubygems, python packet manager en perl packet manager. Hiermee kun je snel verschillende pakketten installeren en aangeven welke pakketten je applicatie nodig heeft zodat deze vanzelf geïnstalleerd worden als vereiste voor je applicatie.

Het lijkt erop dat Node.js na twee jaar een zeer goed alternatief aan het worden is voor bestaande serverapplicaties

die gebruik maken van een zeer hoge taal, zoals php, ruby of python. De applicaties die gemaakt worden met Node.js worden steeds geavanceerder. Een voorbeeld van een grote speler die gebruik maakt van Node.js is LinkedIn, die Node.js gebruikt voor hun mobiele applicatie. Volgens Kiran Prasad, hoofd mobiele ontwikkeling bij LinkedIn, is hun applicatie tien keer sneller dan voorheen en is het ontwikkelen ook supersnel. Voorheen gebruikten ze Ruby on Rails.

"Denk jij dat je 14 computers verslaat in schaak?"

Een evenement waar je echt kunt zien wat Node.js kan, is Node.js Knockout. Node.js Knockout werd gehouden van 27 augustus tot en met 29 augustus. 294 teams bestaande uit 720 deelnemers hebben 178 Node.js applicaties ontwikkeld. Chess@home is een voorbeeld van zo'n applicatie. De makers ervan hebben zich als doel gesteld om het wereldrecord grootste schaak AI te breken. Ze zijn van plan een D-day te plannen waarop ze zoveel mogelijk computers nodig hebben die hun Node.js programma draaien en deze te laten spelen tegen een schaakgrootmeester.

Op de site kun je tegen computers spelen die momenteel online zijn. Momenteel is de AI nog niet zo sterk. Met een klein beetje moeite is deze te verslaan. Wat daarbij meespeelt is dat er meestal niet zoveel computers online zijn om tegen te spelen, waardoor de AI zwak is. Om deze AI te helpen heb ik bij mijn test een aantal keer meegeholpen met de AI. Dit kan op twee manieren.

De eerste manier is door het Chess@home-worker Node.js pakket te installeren met de Node.js packet manager. Als je deze dan opstart als Node.js server dan begint het programma output te genereren over schaakspellen die hij op dit moment aan het berekenen is voor Chess@home. Het ziet er allemaal zeer indrukwekkend uit, maar het heeft totaal geen invloed op een computersysteem. Op mijn dual core 2,55 GHz bleef het percentage CPU dat Node.js gebruikte nagenoeg nul. De kracht van Chess@home moet dus echt komen van zoveel mogelijk computers die berekenen.

De tweede manier om je computer uit te lenen om de AI sterker te maken, is naar de website gaan van Chess@home. Ze hebben namelijk een widget op hun site die ook meehelpt de AI sterker te maken. De code van deze widget is nagenoeg hetzelfde als de code van het Node.js pakket, het verschil is dat de widget gebruik maakt van WebSockets en als terugval AJAX in plaats van de in Node.js ingebouwde socket API's.

Dat brengt me terug bij de voordelen van Node.js, omdat je bij Node.js gebruikt maakt van javascript kun je heel veel code van je serverzijde gebruiken in je cliënt en andersom. Het enige verschil zit hem in welke API's je gebruikt. In het voorbeeld hierboven, de WebSockets tegenover de Node.js socket API. Omdat deze twee delen zijn weg geabstraheerd kan de AI gedeeld worden en kunnen bugs die opgelost worden gedeeld worden tussen de twee delen. Dit en de andere bovengenoemde voordelen maken Node.js aantrekkelijk om te gebruiken in het maken van webapplicaties.

Concluderend is Node.js dus een zeer aantrekkelijk alternatief voor bestaande thread-based webservers zoals Apache. Er zijn al zeer interessante applicaties mee gemaakt, zoals Chess@home en de achterkant van de mobiele applicatie van LinkedIn.

Bronnen

Google Trends: "node.js"
http://bit.ly/s1hy6Y

Node.js Interview: 4 Questions with Creator Ryan Dahl
http://bit.ly/e18Wdl

NodeJS About
http://bit.ly/3mw7ov

Event-driven Programming for Robust Software
http://bit.ly/sLZ9OE

Threads Without the Pain
http://bit.ly/ryfqk9

Exclusive: How LinkedIn used Node.js and HTML5 to build a better, faster app
http://bit.ly/nJ7xx8

Node Knockout About
http://bit.ly/pOKzA8

Chess@home
http://bit.ly/mQtsFJ

De waarde van “web 2.0”


Michel Brinkhuis
Redacteur I/O Vivat

DOTCOM BUBBLE, TECHNOLOGIE, BEURS, AANDELEN, WAARDE, OMZET, GROUPON, GOOGLE, FACEBOOK, HANDEL

Een nieuwe dotcom-bubble?

Je leest er de laatste tijd steeds meer over: de beurswaarde van bedrijven zoals Facebook en Twitter. Steeds meer ‘internetbedrijven’ overwegen de stap naar de beurs, of zijn al in een serieus stadium beursgenoteerd te raken. Er zijn echter ook critici die terug denken aan een situatie iets meer dan tien jaar geleden, namelijk de ‘dotcom-bubble’. Het feit dat op de Wikipedia-pagina over een beursgang in het algemeen een hoofdstuk ‘internetzeepbel’ is opgenomen, zegt veel over de impact die die ‘bubble’ in het verleden heeft gehad. Ook nu plaatsen veel mensen hun vraagtekens bij de waardering van social-networksites en consorten. Zo uitte de beroemde investeerder Warren Buffet al eerder openlijk zijn twijfels over de waardering van bedrijven in deze sector. Volgens Buffet zullen de meeste bedrijven `overpriced` zijn, omdat het lastig is de werkelijke waarde van een dergelijke onderneming te bepalen. In dit artikel kijken we naar tien bedrijven uit dezelfde sector en hoe hun waarde momenteel is vastgesteld.

Hoe gaat een beursgang?

Wanneer je wat leest over een aanstaande beursgang van een bedrijf, dan spreekt men meestal over een IPO: een ‘initial public offering’. Dat bedrag geeft de prijs aan van de aandelen die op de markt worden gebracht.

Facebook

Facebook is misschien wel het bedrijf dat het meest in het nieuws komt, zowel als het gaat om de waarde van de onderneming als om hoe privacy bij Facebook meespeelt. Momenteel wordt de waarde van het bedrijf op een bedrag tussen de 50 en 70 miljard dollar geschat. De website is zo waardevol omdat het meer dan 600 miljoen actieve gebruikers heeft, waarvan de helft dagelijks inlogt. Elke actieve gebruiker bevindt zich op een dag gemiddeld 55 minuten op Facebook. Dat maakt Facebook met name interessant voor adverteerders.

> ‘Facebook gaat eind 2012 naar de beurs’

Adverteren op Facebook is interessant voor bedrijven, omdat er heel specifiek kan worden ‘getarget’. Als adverteerder is het mogelijk om te zeggen dat je in een bepaald deel van een land wilt adverteren bij alle mannen in de leeftijd van 20 tot 40 jaar. Voor een bedrijf dus een ideaal beeld: adverteren voor enkel je doelgroep. Mocht je willen adverteren op bijvoorbeeld Nu.nl, dan wordt dat een stuk lastiger. Nu.nl heeft immers niet de beschikking over zulke gedetailleerde informatie van haar gebruikers.

Facebook is van plan waarschijnlijk eind 2012 naar de beurs te gaan. Het bedrijf heeft echter onlangs haar 500e aandeelhouder investeerder verwelkomt, en daarmee is Facebook verplicht om in april 2012 financiële cijfers te presenteren. Daarmee zal ook de werkelijke waarde van de sociale netwerksite een stuk duidelijker worden. Op het moment dat Facebook naar de beurs gaat kan iedereen aandelen kopen, op dit moment kun je alleen investeren in overleg met het bedrijf.

Twitter

Microblogdienst Twitter is volgens diverse websites ‘stilletjes’ een beursgang aan het voorbereiden, omdat er onlangs diverse wisselingen in het management achter de dienst plaatsvonden. Hoewel er soms op Twitter ‘promoted Tweets’ te zien zijn, is de website nog steeds zoekende naar een verdienmodel. Zolang er nog geen passend model gevonden is, is de website verliesgevend en afhankelijk van investeerders. De CEO van Twitter, Dick Costello, is van mening dat Twitter vandaag de dag 8 miljard dollar waard is. Het platform heeft meer dan honderd miljoen gebruikers, die samen meer dan een kwart miljard keer tweeten op een dag. In oktober dit jaar gaf de Twitter-topman aan dat het bedrijf zelfstandig wil blijven, en voorlopig nog geen beursgang overweegt.

Skype

Een paar maanden geleden is het voipprogramma Skype overgenomen door Microsoft. Het bedrijf uit Redmond betaalde daar 8,5 miljard dollar voor. Skype heeft momenteel rond de 170 miljoen gebruikers. Ongeveer één procent van deze gebruikers is bereid te betalen voor telefoongesprekken. Zoveel gebruikers, en gegevens over die gebruikers, is natuurlijk zeer interessant voor adverteerders. Skype zal worden gecombineerd

in het Office-pakket van Microsoft, zodat het bedrijf beter de concurrentie met Apple (vanwege Facetime) aan kan gaan. Winstgevend is Skype nog steeds niet, vorig jaar werd er een verlies van 7 miljoen (ongeveer één procent van de totale hoeveelheid geld die in het bedrijf omgaat)omzet) gemaakt.

Groupon

Groupon, een website die gebruikers dagelijks locatie-gerelateerde kortingen biedt, moest onlangs haar gerapporteerde omzet halveren. Dit omdat het volgens de Amerikaanse toezichthouders niet toegestaan was het totale bedrag dat klanten betalen, en waarvan dus nog een deel aan de ondernemer die het product verkoopt moet gaan, bij de omzet op te tellen. Groupon bereidt op het moment van schrijven een gang naar de beurs voor, en daarom wordt de boekhouding van het bedrijf gronding gecontroleerd door de Securities and Exchange Commission. Hoewel Groupon slechts drie jaar bestaat, bedraagt de omzet over de eerste helft van 2011 al maarliefst 688 miljoen dollar. De dienst is echter nog steeds verliesgevend. Wanneer Groupon de beurs betreedt, zal het naar verwachting een waarde van 20 miljard dollar hebben. Eerder dit jaar sloeg de kortingswebsite een overnamebod van 6 miljard door Google af. De vraag is dus of de waarde van tientallen miljoenen terecht is, gezien het feit dat er op dit moment nog geen winstgevend bedrijf aan ten grondslag ligt, en ook analisten het er niet over eens zijn of Groupon dat ooit zal worden.

Google

Eén van de bedrijven waarbij het geen vraag meer is of er überhaupt winst gemaakt kan worden is Google. Inmiddels ook beurgenoteerd, en al vele jaren uiterst winstgevend. Over het afgelopen kwartaal van dit jaar (Q3) kwam de winst van de zoekgigant, die de inkomsten met name uit advertenties haalt, uit op 2,7 miljard dollar. De marktwaarde van Google, die in tegenstelling tot veel andere internetbedrijven wel realistisch kan worden vastgesteld (op basis van de prijs van de aandelen, en het aantal uitgegeven aandelen), nadert de tweehonderd miljard dollar. Het bedrijf heeft voor 47 miljard aan cash en kortetermijnsbezittingen, datzelfde geldt voor IBM en Apple. Alleen Microsoft heeft meer geld in z'n bezit, ongeveer 75 miljard dollar.

LinkedIn

De meer zakelijk georiënteerde sociale netwerksite LinkedIn is een paar maanden geleden naar de beurs gegaan. Voor LinkedIn was dat een succesvolle stap, de waarde van het bedrijf bleek een stuk hoger te liggen dan analisten vooraf hadden verwacht. LinkedIn is de

Google's marktwaarde kan realistisch worden bepaald

Waarom gaat een bedrijf naar de beurs?

Bedrijven gaan naar de beurs om geld aan te trekken. Dat doen ze wanneer ze bijvoorbeeld van plan zijn om andere bedrijven over te nemen: daarvoor is immers veel geld nodig.

Geld zou een bedrijf ook van een bank kunnen lenen, verschil is echter dat bij een bank een vaste rente betaald moet worden. Dat is op de aandelenbeurs niet zo: mensen kopen aandelen in een bedrijf, in de hoop dat de waarde van het aandeel zal stijgen. Die waarde stijgt bijvoorbeeld wanneer het goed gaat met een bedrijf, want dan is er veel vraag naar aandelen.

Ook kan het voor een bedrijf lastig zijn om veel geld te lenen van een bank, voor de bank is het namelijk best een groot risico wanneer ze héél veel geld uitlenen aan slechts één bedrijf. Als belegger leen je in feite maar een relatief klein bedrag uit, en is het risico veel lager. Immers: het bedrag wordt over duizenden beleggers gespreid.

Beleggers beleggen om rendement te behalen: als een bedrijf winst maakt keert het dividend (een deel van de winst) uit aan de aandeelhouders.

eerste social mediasite die de stap naar de beurs heeft gemaakt. Velen zien de beursgang van LinkedIn als een voorproefje voor wanneer Facebook naar de beurs gaat. In het tweede kwartaal van 2011 had LinkedIn een omzet van 121 miljoen dollar, en maakte men 3,1 miljoen euro winst. LinkedIn wil echter meer inkomsten genereren, en is daarom van plan om gebruikers meer naar de website te krijgen, en ook in meer landen dan waar men op dit moment actief is actief te worden.

Baidu

In tegenstelling tot alle andere bedrijven in dit overzicht is Baidu geen westers bedrijf, en ook niet bijster bekend in Europa en de VS. Baidu is een Chinese zoekmachine, en doet in China in feite hetgeen Google hier in het westen doet. In april van dit jaar sloot Baidu een deal met Facebook, om samen de Chinese markt te bestormen met een geheel nieuwe sociale netwerksite. Sinds 2005 staat de zoekmachine al op de NASDAQ, en is dus al opvallend lang aan een beurs genoteerd. Met een omzet van meer dan een miljard dollar in 2010, en meer dan tienduizend werknemers, is Baidu een imposante onderneming. Baidu heeft meer dan 75% van de Chinese 'zoekmarkt' in handen.

Yahoo

Yahoo is net als Google een beursgenoteerd bedrijf. Tevens is het een bedrijf met een heel wat langere historie dan de meeste bedrijven in deze top 10. In 2008 probeerde Microsoft met een bod van zo'n 50 miljard dollar Yahoo over te nemen. Yahoo sloeg het bod af, want men vond het te laag. Misschien een verkeerde beslissing, want de waarde van Yahoo is de afgelopen jaren flink gedaald. Vandaag de dag is het bedrijf, met name bekend van de gelijknamige zoekmachine, iets minder dan 20 miljard dollar waard. Microsoft zou, zo werd begin Oktober bekend, zoeken naar een partner om opnieuw een overnamebod op Yahoo te doen. Duidelijk mag zijn dat de waarde van het bedrijf in een paar jaar tijd meer dan gehalveerd is. Winstgevend is Yahoo nog altijd, zo bedroeg de winst over het tweede kwartaal van dit jaar 213 miljoen dollar, bij een omzet van 1,6 miljard dollar.

Zynga

Zynga is misschien één van de minder bekende bedrijven in dit lijstje. Het bedrijf maakt spelletjes, en is bekend geworden dankzij met name het op Facebook populaire spel FarmVille. Inmiddels zijn er heel wat varianten, zoals FrontierVille en CityVille, die het bedrijf ook de nodige miljoenen euro's omzet opleveren. Het verdienmodel van Zynga is gebaseerd op 'in game' aankopen. Dagelijks zijn er 60 miljoen actieve spelers, en wordt er twee miljard uur per dag gespeeld. Dat laatste zorgt voor vier miljard 'in game user-to-user connections'. De omzet in 2010 bedroeg 597,5 miljoen dollar. In vergelijking tot een omzet van 19,4 miljoen twee jaar eerder, kan worden aangenomen dat het bedrijf een enorme groei doormaakt.

MySpace

Wellicht een vreemde eend in de bijt in deze lijst. Immers, MySpace lijkt aardig over z'n hoogtijdagen heen te zijn, en de waarde lijkt met de dag minder te worden in plaats van de stijgen. Dit alles, aldus experts, dankzij de opmars van Facebook. De waarde van MySpace bedraagt op dit moment 35 miljoen dollar, getuige de geschatte waarde van de overname door het bedrijf Specific Media. Daarmee is MySpace eigenlijk een vreemde eend in de bijt wat betreft dit overzicht: de waarde ligt een stuk lager dan alle andere genoemde bedrijven, maar MySpace is minstens zo bekend.

De toekomst

Het is natuurlijk lastig te zeggen of er een nieuwe dotcom-bubble aankomt. De situatie is op dit moment behoorlijk anders dan vijftien jaar geleden. Iedereen beschikt nu namelijk over een razendsnelle internetverbinding, en een groot deel van de mensen is ook mobiel niet onvoorzien van een internetverbinding. Onze afhankelijk van het internet is enorm toegenomen, en ook de plaats die internet en haar toepassingen inneemt in het huidige maatschappelijke klimaat is in bijna niets met wat het ten tijde van de dotcom-bubble was.

Bronnen

Facebook puts off IPO until 2012
Financial Times http://on.ft.com/rZkdQV

Social Network sites are overpriced
Bloomberg, http://bloom.bg/hy3vke

Omzet Groupon gehalveerd
Nu.nl, http://bit.ly/oOTTUN

How much cash do large tech-companies have?
Tecparatodos.com, http://bit.ly/ut4Evy

Advertentie Technolution

De veiligheid van RSA


Rick van Galen
Redacteur I/O Vivat

RSA, CRYPTOGRAFIE, QUANTUM-COMPUTERS,PRIEMFACTORISATIE, P VS NP

Hoe veilig is veilig op het internet?

Bij de meerdereheid van digitale transacties wordt vertrouwd op de encryptie en authenticatie die RSA biedt. Over het algemeen wordt het cryptosysteem als veilig beschouwd. Maar waar is deze veiligheid op gebaseerd? En wat zijn de gevaren?

De invoering van RSA door de cryptologen Rivest, Shamir en Adleman was het eerste cryptosysteem dat asymmetrisch was. Traditionele encryptie, zoals het Amerikaanse DES en de Enigma-machine uit de Tweede Wereldoorlog, gaan ervan uit dat een bericht wordt versleuteld en ontcijferd met dezelfde sleutel. Dit is bij RSA niet het geval. Bij RSA is er sprake van een zender en ontvanger. De zender genereert twee sleutels, de public en private key. De public key wordt uitgezonden, zodat iedereen een bericht kan versleutelen. Met de private key kan de uitzender dit bericht weer decoderen.

Het systeem van RSA is te vergelijken met een brievenbus: iedereen kan er een brief in doen, maar alleen de postbode heeft een sleutel om bij de berichten te kunnen. Dit eenwegsprincipe is interessant omdat het mogelijk maakt voor een grote groep gebruikers veilig informatie centraal op te slaan. Ook biedt de uniekheid van de partij met de privesleutel mogelijkheden tot authenticatie.

Deze authenticatiemogelijkheden staan aan de basis van de certificaten die gebruikt worden bij informatie-uitwisseling op het web. De certificaten die bijvoorbeeld worden gebruikt om de verbinding naar betalingsinstanties als PayPal of banken worden gebruikt. De HTTPS-verbinding maakt gebruik van een veilige verbinding waar het certificaat van de instantie over wordt verstuurd. De privesleutel van de instantie wordt gebruikt om het certificaat, alsmede een sleutel van een certificate authority, om het certificaat te authenticeren. De certificate authority is hierbij een instantie die verifieert dat het uitgegeven certificaat inderdaad van de organisatie is die het zegt te zijn. De veiligheid van deze certificaten hangt af van de integriteit van deze certificate authority, en de veiligheid van RSA. Op deze laatste veiligheid gaan we in in dit artikel.

Priemfactorisatie in polynomiale tijd?

Wiskundige werking

De werking van RSA is een wiskundige truuk die het niet onmogelijk, maar bijzonder onpraktisch maakt om de verbinding te ontcijferen. Om een RSA-verbinding op te zetten kiest men twee grote priemgetallen, p en q. De publieke sleutel bestaat dan uit twee getallen, n, dat het product is van p en q, en e, een getal dat gekozen wordt dat copriem is, en kleiner dan (p-1)(q-1). De privesleutel wordt gevonden door het getal te vinden waar voor geldt dat de=1 mod (p-1)(q-1). Dit wordt de inverse van genoemd.

Stel Alice stuurt Bob de publieke sleutel, en Bob wil een bericht versleutelen en naar haar toesturen. Bob zet dan dit bericht om in een getal, zodat hij er mee kan rekenen. Het versleutelde bericht c dat Bob stuurt is dan $c=m^e \bmod n$. Alice kan dit bericht ontcijferen door haar privesleutel te gebruiken. Ze rekent dan uit $m=c^d \bmod n$. Omdat d de inverse van e is wordt machtsverheffing van Bob weer teniet gedaan, en heeft Alice het oorspronkelijke bericht weer teruggevonden.

Als je deze algoritmes bestudeerd, kun je zien dat een aanvaller die de pakketten tussen Alice en Bob onderschept mogelijk de gecodeerde berichten van Bob terug kan halen door de privesleutel van Alice te berekenen. Aangezien Alice de getallen n en e vrijgeeft, kan Bob de getallen p en q afleiden en simpelweg de privesleutel bereken door de=1 mod (p-1)(q-1) op te lossen. De veiligheid van RSA zit er echter in dat uit het getal n niet zomaar de getallen p en q af te leiden zijn. Om deze af te leiden moet n gefactoriseerd worden in de priemgetallen p en q, en dat is niet op een snelle manier te doen, zeker niet als de getallen heel groot zijn. In de meeste implementaties van het RSA-algoritmes hebben de getallen een lengte van tenminste 1024 bits.

Priemfactorisatie

Er zijn wel een aantal algoritmen die priemfactorisatie kunnen uitvoeren - het onderwerp wordt veel onderzocht.

De snelste algoritmen hebben een draaitijd die exponentieel is met de lengte van het te factoriseren getal in bits. Het getal dat gefactoriseerd moet worden in RSA heeft een grootte van tenminste 1024 bits, en meestal 2048 of 3072 bits.

Gerust hart de komende jaren. Waarschijnlijk.

De grootte van deze getallen maakt het onpraktisch om er deze algoritmen op los te laten. Computers zouden jaren moeten rekenen om de gewenste factorisatie te berekenen van zelfs de kortste lengte. De veiligheid van RSA is dus gegarandeerd door de praktische onmogelijkheid van het berekenen van de factorisatie.

Er kleven echter nog wel wat theoretische problemen aan. Hoewel de algoritmen praktisch niet effectief zijn in het factoriseren van de priemgetallen, is het nog niet aangetoond dat deze priemfactorisatie wel degelijk zo'n lastig probleem is. Van een groot aantal algoritmen is bepaald of zij in de complexiteitsklasse P of niet liggen - de complexiteitsklasse waarin algoritmen zitten die in polynomiale tijd zijn op te lossen. Hoewel de veiligheid van RSA juist ligt in het feit dat het factorisatie-algoritme niet in polynomiale tijd draait, is het nog nooit aangetoond dat het probleem in een complexiteitsklasse zit die dit zou garanderen. Als oplossing voor dit probleem zijn er cryptosystemen voorgesteld waarvan het zeker is dat deze in NP liggen, zoals een cryptosysteem op basis van het handelsreizigersprobleem. Mocht het factorisatieprobleem alsnog de complexiteitsklasse NP toegewezen krijgen, dan nog zal het niet duidelijk zijn of RSA veilig is. Dit is omdat de relaties tussen complexiteitsklassen nog niet duidelijk zijn: een bekend probleem is de vraag of de klasse van polynomiaal oplosbare problemen P, en de klasse van de problemen waarvan een gegeven oplossing polynomiaal is na te gaan NP, gelijk zijn aan elkaar.

Het is dus niet onmogelijk (of, nog niet aangetoond) om een algoritme te vinden dat deze priemfactorisatie in polynomiale tijd mogelijk maakt. Een invalshoek die mogelijk werkt loopt via een ander bekend probleem in de wiskunde: de Riemann-hypothese. De Riemann-hypothese is een stelling door de 19e-eeuwse wiskundige Bernhard Riemann, die bij bewijs gevolgen zou hebben voor de voorspelbaarheid van de locatie van priemgetallen op getallenlijn. Hoewel de Riemann-hypothese al meer dan 150 jaar onbewezen is, en er nog geen methode is ontwikkeld die op basis van de waarheid van de hypothese priemgetallen kan factoriseren, moeten de ontwikkelingen omtrent deze goed in de gaten worden gehouden door cryptologen.

Quantumcomputers

Voor gewone computers is er geen algoritme gevonden dat de priemfactorisatie van RSA in polynomiale tijd op kan lossen. Een dergelijk algoritme is echter al wel beschreven en ook getest voor quantumcomputers. De mogelijkheid om in plaats van bits te werken met qubits biedt de mogelijkheid om een aantal optimalisaties te maken die het algoritme vele malen sneller kunnen lopen. Qubits in quantumcomputers onderscheiden zich van gewone bits in het feit dat zij niet slechts de waarde 0 of 1 kunnen aannemen, maar in het bijzonder een superpositie van 0,1 of beide in meer of mindere mate.

Deze eigenschap maakt het mogelijk om bepaalde operaties op informatie veel sneller uit te voeren dan mogelijk is met conventionele bits. Het algoritme van Shor is een algoritme dat beschrijft hoe op quantumcomputers een arbitrair getal kan worden gefactoriseerd in zijn priemgetalvorm. Het algoritme heeft de vorm van een 'gewoon' algoritme, met een enkele stap die quantumversnelling vereist. Het idee van het algoritme is het zoeken van de orde van iedere kandidaatfactor in de verzameling getallen kleiner dan het getal dat gefactoriseerd wordt. Er is een stelling die bewijst dat met deze orde een bepaalde vorm heeft, dat deze een priemfactor van het te factoriseren getal is. Door het gebruik van qubits kunnen er veel getallen tegelijk in superpositie worden getest of zij de orde van het getal representeren, en vormen hierdoor een efficient mechanisme om het algoritme op uit te voeren.

Omdat dit algoritme alle integers test die kandidaten zijn voor een priemfactor van het te factoriseren getal, is de looptijd van het algoritme in de orde $O(n^{1/2})$: niet alleen in polynomiale tijd

maar ook nog behoorlijk snel.

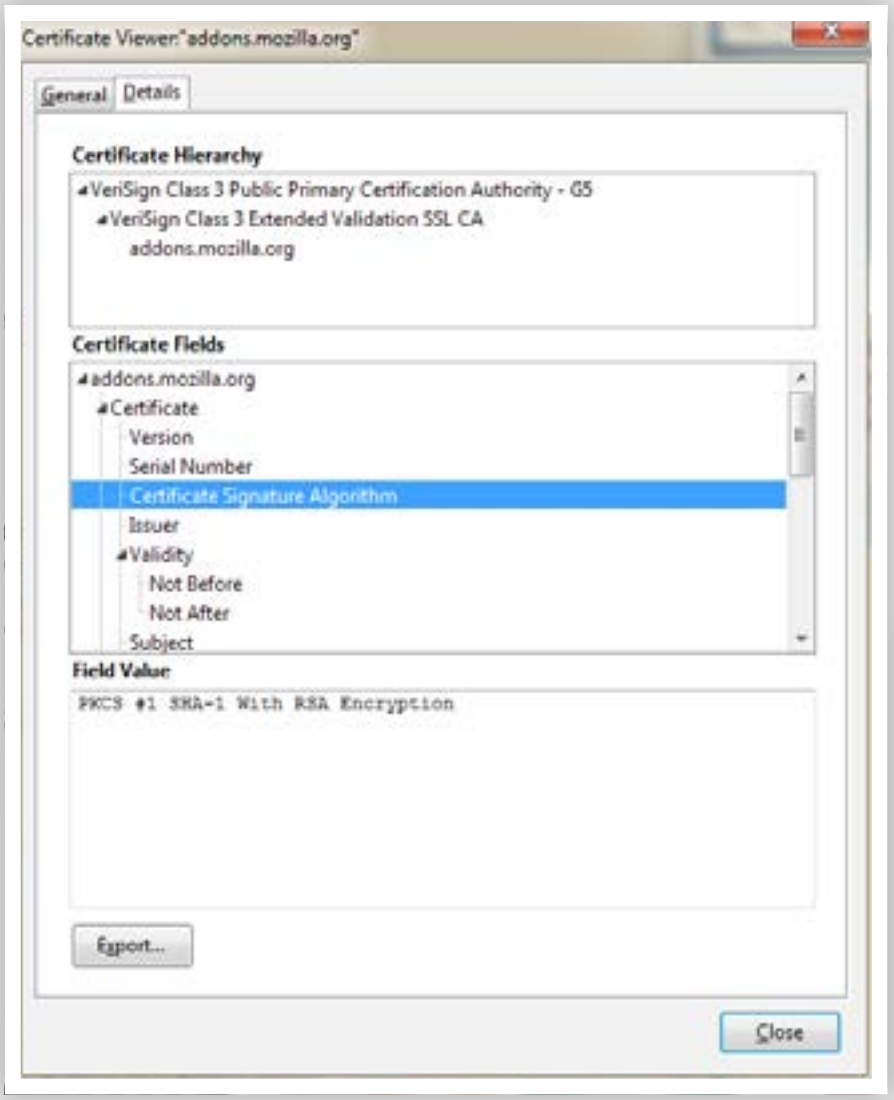
Is RSA hierdoor onveilig geworden? Dat is niet zo. Quantumcomputers zijn bij lange na niet zo ver ontwikkeld dat deze ingezet kunnen worden om getallen te factoriseren in de orde van grootte van de getallen die bij RSA worden gebruikt. Hoe groter het getal, hoe meer qubits er nodig zijn. Sinds 2001 is het algoritme van Shor een aantal keer ingezet om experimentele quantumcomputers te testen, maar geen van deze had de capaciteit om een getal groter dan 15 te kunnen factoriseren.

Vingers gekruisd?

Het lijkt opmerkelijk dat een cryptosysteem met een dergelijke wiskundige basis als RSA zo weinig garanties kan geven over de veiligheid ervan. De veiligheid van RSA is puur afhankelijk van het feit dat de praktijk het algoritme nog niet kan kraken: er is nog geen effectief klassiek factorisatie-algoritme gevonden en er zijn nog geen bruikbare quantumcomputers. Gezien de afhankelijkheid van veel gegevensuitwisseling van dit cryptosysteem kunnen er vraagtekens gezet worden bij het vertrouwen dat in het systeem wordt geuit.

De alternatieve wijdverspreide cryptosystemen zijn het Elgamal-systeem en het elliptische-kromme-systeem. Voor deze algoritmen valt echter ook bovenstaande redenatie toe te passen - hoewel de praktijk er nog niet klaar voor is, zijn er geen garanties dat dit in de toekomst niet zo gaat zijn.

De keuze voor RSA is daarom niet raar. Bovendien zijn de praktische kwesties die opgelost moeten worden om RSA te kraken niet triviaal. We kunnen de komende jaren onze internetbetalingen en e-mailauthenticatie daarom nog met een gerust hart uitvoeren. Waarschijnlijk.



Figuur 1: Certificaatinformatie in Firefox.

Bronnen

Cryptopsysteem op basis van handelsreizigersprobleem
http://dl.acm.org/citation.cfm?id=1022392

P vs NP
http://www.claymath.org/millennium/P_vs_NP/

De Riemann-hypothese
http://mathworld.wolfram.com/RiemannHypothesis.html

Het algoritme van Shor
http://alumni.imsa.edu/~matth/quant/299/paper/node21.html

Istanbul

Rom Langerak
Opleidingsdirecteur Informatica

Sinds april 1992 is dr. ir. Rom Langerak universitair docent bij de Formal Methods and Tools groep van de faculteit EWI. Romanus (Rom) werd op 1 februari geboren in Dordrecht en ging naar het Christelijk Lyceum aldaar. Hij haalde op de Universiteit Twente met lof zijn studie Toegepaste Wiskunde, waar hij afstudeerde op een onderwerp over Databases. Het is dan ook niet vreemd dat hij na zijn afstuderen ging promoveren bij de toenmalige faculteit Informatica. Na zijn promoveren in 1992 bleef hij bij de faculteit werkzaam.

Rom houdt van literatuur, filosofie, gitaar spelen, biljarten en Taekwondo. Sinds september 2009 is hij de nieuwe opleidingsdirecteur Informatica, een taak die hij met liefde zal gaan uitvoeren om zo het onderwijs voor zowel studenten als docenten

Een paar maanden geleden werd ik benaderd door een charmante dame van Strategie en Beleid met de vraag of ik zin had mee te gaan naar Turkije. Turkije is een van de landen waar de UT zich op richt bij de werving van internationale studenten, en men probeert de opleidingsdirecteuren daar wat meer bij te betrekken. Op het programma stonden bezoeken aan diverse universiteiten, een etentje met Turkse UT-alumni, en een tweedaagse beurs waarop universiteiten uit de hele wereld zich presenteren aan Turkse potentiële studenten.

Wat wist ik op dat moment van Turkije? Eigenlijk niet veel. Een onderbuurman in mijn flat heeft een Turkse eettent in Hengelo. Op de gang op kantoor hebben we professor Mehmet Aksit, met in zijn kielzog studenten, PhD's en Postdocs uit Turkije, zodat baklava en raki geen onbekende fenomenen zijn. En verder heb ik wat boeken gelezen van literaire Nobelprijswinnaar Orhan Pamuk (vooral zijn autobiografische "Istanbul" is een aanrader). Ik was nog nooit in Turkije geweest, dus ik aarzelde niet lang en zei ja.

Ik ben nu net terug van vijf dagen Istanbul en het was een geweldige ervaring. Istanbul is een wereldstad met tussen de 15 en 20 miljoen inwoners. Het heeft een imposant verleden als hoofdstad van achtereenvolgens het Romeinse, Byzantijnse, en Ottomaanse rijk. Het bestaat uit een Europees en een Aziatisch gedeelte, gescheiden door de Bosporus , die vanuit de op heuvels gebouwde stad steeds weer onverwachts te zien is. Het combineert hectiek (hordes gele taxis in de vaak stapvoets voortbewegende verkeerschaos) met pittoreske en exotische charmes. Aan de ene kant zijn er veel sprookjesachtige moskeeën met slanke minaretten, aan de andere kant zag ik minder hoofddoekjes dan in bijvoorbeeld Amsterdam.

Politiek is Turkije een boeiend land (islam in een seculaire staat), hoewel zeker niet zonder problemen en spanningen. En economisch loopt het als een trein. Ik denk dat de Turken inmiddels erg blij zijn dat het in 2004 niet lukte om tot de EU toe te treden! Met name high tech industrie is volop in ontwikkeling. Er zijn nu ruim 70 miljoen Turken, en een groot gedeelte daarvan is jong en op zoek naar een goede universitaire opleiding. En hebben steeds vaker ouders die bereid zijn daarvoor te betalen! Vandaar dat de particuliere universiteiten als paddenstoelen uit de grond schieten.

Kijk je naar de informaticaopleidingen daar, dan zie je dat die soms gerund worden door niet meer dan een staflid of tien, waarbij elk staflid in zijn eentje een gebied bestrijkt dat bij ons door een hele leerstoel wordt afgedekt. Voor de bachelor is dat tot daar aan toe, maar voor de master leidt dat tot een opleiding die diepte mist en in vergelijking met onze Twentse informaticamasters veel oppervlakkiger is. Wij geven misschien in de master duur onderwijs (veel vakken voor relatief weinig studenten), maar dat leidt er wel toe dat onze masteropleidingen uitstekend kunnen concurreren met buitenlandse masters als de Turkse. Zou daar niet een geweldige business opportunity in zitten?

En over business opportunities gesproken: wij kennen in Nederland allerlei Turks fastfood, zoals dürum, döner en lacmahun. Maar om onbegrijpelijke redenen is kumpir (spreek uit: koempier) hier volledig onbekend. Het gaat om een king size gepofte aardappel, opengesneden en gemengd met boter en kaas, en vervolgens naar wens van de klant gevuld met allerlei soorten groente en/of vlees. Begin hier een kumpir kiosk, bijvoorbeeld bij de Muur, en je bent binnen!

Beveiligings-certificaten


Niek Tax
Redacteur I/O Vivat

INTERNET, BEVEILIGINGSCERTIFICATEN, PUBLIC KEY INFRASTRUCTURE

Hoe veilig zijn beveiligingscertificaten?

Voor websites waarop de gebruiker vetrouwelijke informatie opgeeft zoals een bank- of overheidswebsite is het van groot belang dat de webbrowser van de gebruiker verbinding maakt met de bedoelde website. Met het bestaan van vervalsingen van met name bankwebsites met phishing als doel ligt gevaar hier op de loer.

Voordat een website online wordt gebracht genereert de eigenaar van de website een public-private keypaar. Alle informatie welke met de private key wordt versleuteld kan worden ontsleuteld met de bijbehorende public key en andersom. De private key houdt de eigenaar voor zichzelf. Voor de public key vraagt de eigenaar aan een zogenaamde certificaatautoriteit om een garantie af te geven dat het public key daadwerkelijk toebehoort aan de hostname. De certificaatautoriteit doet dit door de public key samen met de hostname in een digitaal certificaat te stoppen en namens de certificaatautoriteit digitaal te ondertekenen ter goedkeuring. Hiermee garandeert de certificaatautoriteit dat de aanbieder van het public key de rechtmatige eigenaar van de betreffende website is.

Veel geuit kritiek op het certificatensysteem is het feit dat...

Nu de public key van de eigenaar door een certificaatautoriteit wordt onderkent als "zijn eigendom" kan de eigenaar de key gebruiken voor zijn website.

Figuur 1: Waarschuwing van Firefox wanneer een certificaat niet overeenkomt met het domein waarvoor het is verstrekt.

Zodra een internetgebruiker middels https de website in kwestie bezoekt stuurt de website het certificaat toe dat de certificaatautoriteit hiervoor heeft uitgegeven. Alle pakketten die de bezoeker verstuurd naar de website worden versleuteld met de public key die hij van de website heeft ontvangen. Enkel die partij die de bijbehorende private key in zijn bezit heeft is in staat de boodschap van de bezoeker te decoderen. Dit zorgt ervoor dat de gebruiker er zeker van kan zijn dat hij met de website communiceert waarvan hij het beveiligingscertificaat heeft ontvangen, mits de website-eigenaren hun private key niet laten uitlekken.

Certificaat revocation

Wanneer een derde bezit is gekomen van de private key van een website is het voor deze derde mogelijk geworden zich voor te doen als deze website. 1) Hij heeft namelijk de private key om gegevens versleuteld met de public key van deze website te ontsleutelen. 2) Het certificaat kan hij zelf ook uitgeven, deze verkreeg hij immers al bij het bezoeken van de website. Om te voorkomen dat op deze wijze misbruik gemaakt kan worden zal de site-eigenaar zodra hij merkt dat zijn private key afhandig is gemaakt de certificaatautoriteit inlichten om een nieuwe private key, public key en certificaat te laten maken. De certificaatautoriteit zal het niet meer geldige certificaat op een lijst met ingetrokken certificaten publiceren, de zogenaamde Certificate Revocation List.

Een van de beveiligingsproblemen met het huidige gebruik van beveiligingscertificaten is het feit dat juist dit mecha-

nisme van het intrekken van certificaten niet goed werkt. Om dit mechanisme goed te laten werken zouden webbrowsers en applicaties elke keer voordat een certificaat wordt geraadpleegd bij de uitgever van dit certificaat moeten controleren of dit certificaat nog niet is ingetrokken. Voor het uitvoeren van een dergelijke controle bestaat het Online Certificate Status Protocol (OCSP). Helaas werken de meeste webbrowsers op het moment zo dat een certificaat geaccepteerd wordt wanneer een controle op de geldigheid van een protocol via OCSP lang duurt.

Gevaarlijker wordt het wanneer het een kwaadwillende lukt om binnen te komen bij een certificaatautoriteit, zoals recentelijk bij Diginotar. De locatie waar de geldigheid van een certificaat wordt gecontroleerd staat vermeld in het certificaat zelf, deze locatie staat in het CRL-veld (Certificate Revocation List) van het certificaat. Wanneer de inbreker zelf in staat is certificaten aan te maken kan hij bij de certificaten die hij aanmaakt het CRL-veld instellen op een eigen server. Dit zal tot gevolg dat intrekking van de door de hacker aangemaakte certificaten de gebruikers van deze certificaten niet meer zal helpen, controle op geldigheid gebeurd nu immers op een ander adres welke in bezit is van de hacker.

Veel geuit kritiek op het huidige certificatensysteem is het feit dat alle certificaatautoriteiten in principe overal certificaten voor kunnen uitgeven. Hiermee zou één onbetrouwbare certificaatautoriteit in beginsel de veiligheid op internet kunnen beïnvloeden. De recente inbraak bij Diginotar heeft de discussie weer doen oplaaien. Er zijn vele certificaatautoriteiten waarbij onmogelijk gesteld kan worden dat deze allen hun beveiliging op orde hebben. Toekomstige problemen in navolging van Diginotar en de eerder gehackte certificaatautoriteit Comodo kunnen niet worden uitgesloten. Daarnaast zijn er enkele certificaatautoriteiten, zoals het Chinese China Internet Network Information Center (CNNIC) waarvan twijfel bestaat over hun integriteit. Een Chinese certificaateutoriteit die dicht op de Chinese overheid staat zou de Chinese staat kunnen assisteren in digitale spionage.

... alle certificaatautoriteiten in principe overal certificaten voor kunnen uitgeven

Bronnen

Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure (2000)
http://www.schneier.com/paper-pki.pdf

Web Security Trust Models (2010)
https://freedom-to-tinker.com/blog/sjs/web-security-trust-models

Mozilla Debates Whether to Trust Chinese CA (2010)
https://freedom-to-tinker.com/blog/felten/mozilla-debates-whether-trust-chinese-ca

Browser finger-printing

Ralph Broenink
Redacteur I/O Vivat

DATABASES, XML, SOCIALE WETENSCHAPPEN, OPENBAAR VERVOER, LIJST, MET, ONDERWERPEN, OF, KERNWOORDEN

Zo anoniem surf je niet

We weten allemaal wel dat het moeilijk is om op internet anoniem te blijven. Op vrijwel iedere website waar je komt, is het mogelijk om een account aan te maken, waarbij er natuurlijk ook om een naam, adres en woonplaats wordt gevraagd. Maar zelfs als je geen account aanmaakt, dan word je op het gros van de websites alsnog gevolgd.

Het volgen van gebruikers is zeer belangrijk voor een aantal doeleinden. Allereerst is het voor de advertentienetwerken, die samen een omzet van 26 miljard dollar representeren, van belang om gerichte advertenties en statistieken te verzamelen. Zo kan er voor worden gezorgd dat jij advertenties over de laatste games te zien krijgt en je moeder een kookboek aangeboden krijgt. Ten tweede wordt het gebruikt door website-eigenaren om een beeld te krijgen van het gebruik van de website, zodat deze nog verder geoptimaliseerd kan worden.

De oudste techniek om gebruikers te volgen, is op basis van IP-adres. Voordat na was gedacht over proxies en NAT, was dat een goede indicator voor een 'computer'. Tegenwoordig heb je daar niet zo heel veel meer aan; vaak wordt een IP-adres door meerdere computers gedeeld en wordt een computer door meerdere mensen gebruikt. Bovendien zijn tegenwoordig veel producten mobiel en veranderen deze vaak van IP-adres.

Cookies

Cookies zijn een goede vervanging. Tracking cookies bevatten een unieke identifier voor jouw computer en worden in jouw browser geplaatst. Al jarenlang worden deze door verschillende websites gebruikt om het gedrag van hun bezoekers vast te leggen; een website met tracking cookies is eerder regel dan uitzondering.

Er zijn manieren om je gedrag bij te houden die onopgemerkt zullen blijven

Het nadeel van tracking cookies is dat het gebruik ervan zeer eenvoudig op is te sporen en dat veel middelen bestaan om te voorkomen dat deze worden geplaatst of waarmee ze eenvoudig verwijderd kunnen worden. Iedere browser heeft bijna wel een knop om alle persoonlijke data te wissen en virusscanners hebben vaak definities voor tracking cookies. Als de cookie eenmaal weg is, is deze natuurlijk niet meer te herstellen door de website.

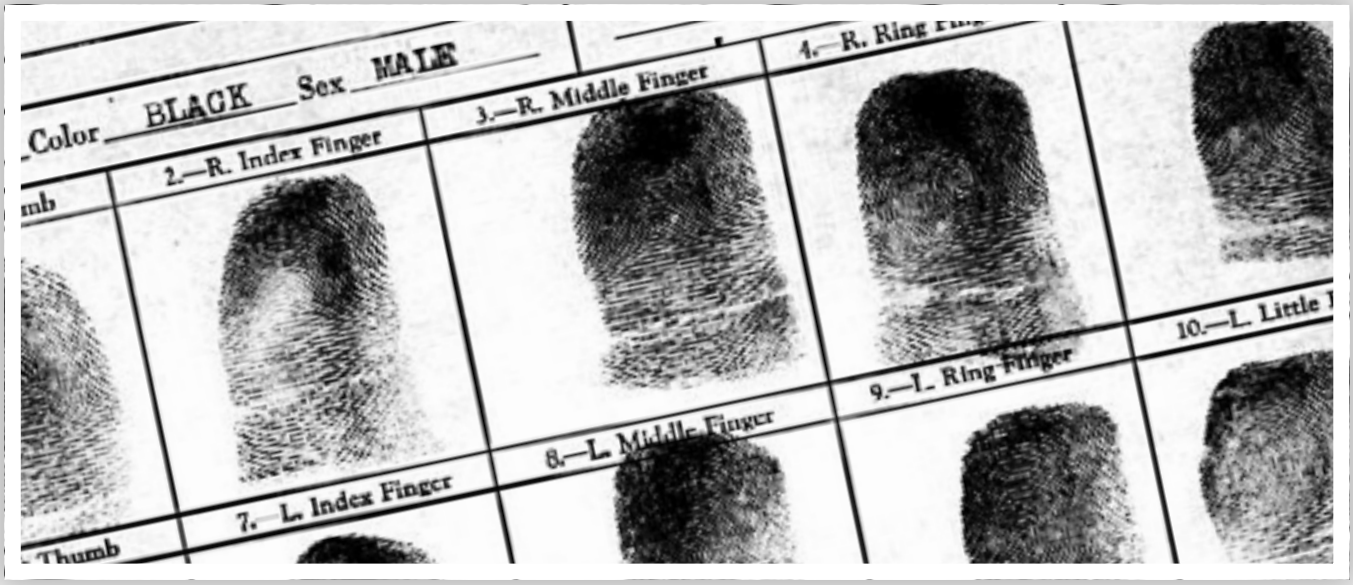
Met 'zombiecookies' (ook wel bekend als 'supercookies' of 'forever cookies') wordt dat laatste probleem grotendeels omzeild. Door het plaatsen van de cookie door middel van Flash ('Local Shared Objects'), Silverlight ('Isolated Storage') of met behulp van HTML5, wordt het verwijderen van cookies een stuk moeilijker en kan een website altijd weer de informatie die in de cookie hoort te staan terughalen en de cookie weer terugplaatsen.

Noemenswaardig is ook het gebruik van HTTP ETags. ETags worden opgeslagen in de cache van de browser en zijn een unieke identifier voor de versie van de website de een browser lokaal heeft opgeslagen. Bij een volgende aanvraag van de browser, wordt de opgeslagen ETag meegestuurd, zodat de website kan beslissen of de versie nog recent is. Omdat de ETag uitgedeeld wordt door de website en door de browser zelf op wordt gestuurd, kan dit uitstekend worden gebruikt om een browser uniek te identificeren. In Amerika loopt al een class-action lawsuit tegen een twintigtal websites die ETags gebruikten, waaronder Spotify en AOL.

De overheid is echter ook wakker geworden en met de nieuwe Telecommunicatiewet moet er ondubbelzinnig toestemming worden gegeven voor het plaatsen van cookies. De nieuwe wet is zo geformuleerd dat ook zombiecookies hieronder kunnen vallen. 'Gelukkig' zijn er nog andere methoden om jouw gedrag bij te houden. Manieren die onopgemerkt zullen blijven en waar je niet heel eenvoudig onderuit kunt.

Fingerprinting

Deze methoden vallen onder het zogenaamde 'browser fingerprinting': het maken van een vingerafdruk van jouw browser. Dat kan vrij eenvoudig met



de informatie die jouw browser ter beschikking stelt aan alle websites die het bezoekt. En dat is een schat aan informatie; kijk alleen al naar de informatie die standaard op wordt gestuurd:

• **User Agent:** Bij iedere aanvraag die je doet, stuurt je browser de 'User Agent' mee. Dit bevat onder andere de versie van je browser, oorspronkelijk bedoeld zodat website-eigenaren kunnen zien door welke browsers de site wordt bezocht en of dat er normaal uitziet. De informatie is echter behoorlijk uitgebreid. Mijn browser stuurt bijvoorbeeld dit mee:

```
Mozilla/5.0 (Windows
NT 6.1; WOW64)
AppleWebKit/535.7
(KHTML, like Gecko)
Chrome/16.0.904.0
Safari/535.7
```

Hier staat dus onder andere dat ik Chrome 16.0.904.0 op een 64-bit-versie (WOW64) van Windows 7 (NT 6.1) gebruik. Daarnaast staan er nog wat andere versienummers, die de eigenschappen van de browser aangeven.

• **Accept:** Bij een aanvraag stuurt je browser ook mee wat voor informatie hij wil zien. Hierbij gaat het om het documentformaat, de encoding, de taal en de karakterset die ondersteund worden:

```
Accept: text/html,
application/xhtml+xml,
application/xml;
q=0.9,*/*; q=0.8

Accept-Encoding:
gzip,deflate,sdch

Accept-Language: en-GB,en;
q=0.8,en-US; q=0.6,nl;
q=0.4

Accept-Charset: ISO-8859-
1,utf-8; q=0.7,*; q=0.3
```

Deze headers zijn grotendeels afhankelijk van de instellingen die je hebt gedaan in je browser. Zo heb ik aangegeven dat ik graag Engelse versies van websites zie en als dat niet kan, dan heb ik graag de Nederlandse.

Een lijst met geïnstalleerde lettertypes uitlezen met behulp van Flash of Java

• **Referer:** De meeste browsers sturen ook nog mee van welke pagina je vandaan komt. Hoewel deze informatie op zich je niet kan identificeren, kan hiermee wel een 'pad' door het internet worden getekend. Als er bijvoorbeeld op twee websites tracking code van hetzelfde bedrijf staat, dan kan er op de eerste website worden gedetecteerd dat jij de pagina verlaat en dat er op de tweede op precies hetzelfde moment iemand vanaf de eerste site binnenkomt.

Met deze informatie zijn we er nog niet, want er zijn natuurlijk duizenden mensen die graag de en-US-versie van een website willen zien en Chrome draaien op een Windows-computer. Javascript kan hier een uitkomst bieden, want onder andere je schermresolutie en het aantal kleuren, de tijdzone van je computer en de taal van je browser kunnen eenvoudig uit worden gelezen.

Ook kan er met Javascript een lijst van plugins op worden gevraagd, met daarbij alle versienummers en ondersteunde mimetypes. Bovendien is deze lijst niet gesorteerd op naam, dus uit slechts de volgorde van deze lijst kan alweer informatie worden verkregen. We kunnen de dataset nog verder uitbreiden door de lijst van geïnstalleerde lettertypes uit te lezen met behulp van Flash of Java. Ook deze lijst is niet gesorteerd op naam; deze lijst is grotendeels uniek per computer gesorteerd.

We kunnen het nóg gekker maken. Ook de lijst van headers die een browser meestuurt per aanvraag, is op een unieke manier gesorteerd: iedere browser kiest zijn eigen volgorde. Verder kunnen we systeeminformatie opvragen met behulp van Silverlight en ActiveX, kan de gebruikersnaam op de computer vaak eenvoudig achterhaald worden door lekken in browsers (denk aan een Javascript-stacktrace die het volledige pad naar de cache geeft) en zijn er verschillende tests om te kijken of een bepaalde extensie is geïnstalleerd (denk hierbij bijvoorbeeld aan een adblocker).

Op een wat lager niveau zijn er nog meer zaken die je browser en je computer identificeren. Ten eerste doet je

browser op een unieke manier aanvragen bij de webserver. Neem een website als google.com. Daar staan een logo, een aantal Javascriptbestanden en een aantal stijlbestanden. Deze moeten bij het bezoeken van de website door de browser op worden gevraagd. Maar iedere browser is vrij om te kiezen hoe dat gebeurt; de extra bestanden kunnen in één keer op worden gevraagd, één voor één of met een maximum aantal per domein. Deze keuzes zijn uniek voor de browser en soms zelfs per versie of per besturingssysteem waar het op draait.

Een andere methode op laag niveau is TCP/IP fingerprinting. Ieder besturingssysteem maakt zijn eigen keuzes voor wat betreft de grootte van het pakket, de TTL (time-to-live), window size en een aantal andere velden. Als laatste is er nog het meten van de clock skew: de hoeveelheid tijd die de klok te snel of te langzaam loopt. Dit kan door middel van TCP en kan resultaten opleveren tot wel 50ms verschil.

Gevaar

De grote vraag is natuurlijk hoe gevaarlijk dit allemaal is. Zijn we echt zo eenvoudig te identificeren? In het Panopticlick-onderzoek van de Electronic Frontier Foundation (EFF) in 2010 is gebleken dat, met slechts een beperkt aantal parameters, iedere browser minimaal 18.1 bits aan identificerende informatie opstuurt. Dat houdt in dat er maar 1 op de 218.1=281.000 browsers dezelfde vingerafdruk heeft. In het onderzoek is gebleken dat 94.2% van de onderzochte browsers een unieke vingerafdruk heeft. Het meten van meer browsereigenschappen en de clock skew (+4-6 bits) en TCP/IP-stack kan enorm bijdragen aan de vingerafdruk die gemaakt kan worden.

Het wijzigen van je vingerafdruk is, in tegenstelling tot je eigen vingerafdruk, vrij eenvoudig. Je hoeft maar een andere voorkeurstaal te kiezen en je bent klaar. Of je updatet één plugin en het versienummer klopt niet meer. In het eerdergenoemde onderzoek is daar ook naar gekeken en (helaas) is in 99,1% van de gevallen een veranderde vingerafdruk goed geïdentificeerd.

Alleen mobiele apparaten (iOS, Android, Blackberry) zijn moeilijk uniek te identificeren: ze worden op grote schaal geproduceerd en er is weinig ruimte om zelf instellingen te doen aan taal en plugins. Ook in omgevingen waar wordt gewerkt met systeemimages, zoals op de universiteit en in verschillende kantoren, is het moeilijk om een juiste identificatie te doen.

Voorkomen

Er is niet aan fingerprinting te ontkomen. Je browser heeft nou eenmaal bepaalde kenmerken en daar ben je nou eenmaal van afhankelijk. Daarnaast is er nog een paradox die meespeelt: als je probeert jezelf te maskeren, val je juist meer op. Als je bijvoorbeeld geen User-Agent meer meestuurt, zal je juist aan het feit dat die er niet staat opvallen. Een directe analogie met een collegezaal is eenvoudig te trekken: als jij de enige bent die een zonnebril op zet, dan kan de docent je toch identificeren; hij kan de kleur van je ogen niet meer zien, maar hij weet wel dat jij het bent.

De meeste browsers hebben tegenwoordig een privacymodus, maar die voorkomen op geen enkele manier fingerprinting. Een goede poging wordt ondernomen door de Tor Browser (ongeveer gelijk aan Firefox met de Torbutton). Meer dan voor anderen is het voor gebruikers van het Tor netwerk belangrijk om anoniem te blijven; journalisten en activisten maken intensief gebruik van dit netwerk, hoewel het natuurlijk ook voor minder legale zaken wordt gebruikt.

De Tor Browser maakt van een heleboel zaken een standaardwaarde, waaronder de Accept-headers en de User Agent. Ook zijn plugins standaard uitgeschakeld, waardoor heel veel informatie al verborgen is. Desalniettemin zijn gebruikers van de Tor Browser zelf wel weer als groep te identificeren en bovendien gebruikt maar 20% van de Tor-gebruikers de Tor Browser.

Je kunt ook je vingerafdruk verkleinen zonder van browser te wisselen. Door het uitschakelen van Javascript, Flash en Java laat je je browser al veel minder informatie lekken, hoewel je dan natuurlijk wel weer identificeerbaar bent aan het feit dat je browser geen Javascript, Flash of Java ondersteunt (of uitgeschakeld heeft). Bovendien gebruiken veel websites Javascript en zul je dus behoorlijk verminkt op het internet rondsurfen. Weinig websites maken echter gebruik van je User Agent en als je browser dit ondersteunt, kies dan een generieke User Agent, zoals Firefox 3.6.0 op Windows XP. Zorg er ook voor dat je voor de documentformaat-, encoding- en karaktersetvoorkeuren de standaardinstelling behoudt en kies ook geen hele lange lijst van voorkeurstalen. Als je echt heel erg wantrouwig bent, is het misschien juist een idee om deze instellingen regelmatig te wijzigen en zo je vingerafdruk inconsistent te maken. Maar of deze tips allemaal écht helpen, is nog maar de vraag.

Al met al surf je waarschijnlijk minder anoniem dan je dacht toen je je cookies uitzette. Je browser stuurt ongewild meer informatie over jou dan je tot nu toe dacht en met Javascript en Flash ben je bijna sowieso uniek. Als je wilt weten hoe anoniem jij écht bent, surf dan naar panopticlick.eff.org. Wat de uitslag ook is, voorlopig zal je moeten leren leven met de advertenties van games en kookboeken.

94.2% van de browsers heeft een unieke vingerafdruk

Bronnen

Panopticlick: How Unique – and Trackable – Is Your Browser?
http://panopticlick.eff.org

Remote physical device fingerprinting (2005)
Tadayoshi Kohno, Andre Broido, kc claffy

Tor HTTP Usage and Information Leakage (2010)
Markus Huber, Martin Mulazzani, Edgar Weippl

Browser Fingerprinting from Coarse Traffic Summaries: Techniques and Implications (2009)
Ting-Fang Yen, Xin Huang, Fabian Monrose, and Michael K. Reiter

Van het ENIAC-bestuur

Johan Noltes
Voorzitter ENIAC

Johan Noltes is voorzitter van ENIAC: de ENschedese Informatica Alumni Club. ENIAC is de alumnivereniging voor oud-studenten Informatica, bedrijfsinformatietechnologie en Telematica aan de Universiteit Twente.

Voor slechts € 5,- per jaar kan je al lid worden van deze club. Je krijgt dan in ieder geval de Vivats die jaarlijks verschijnen (meestal zo'n 4 stuks, maar niet helemaal per kwartaal) en uitnodigingen voor de activiteiten die we organiseren (meestal per mail). Daar mag je dan vervolgens (veelal gratis!) aan deelnemen. En al doe je maar eens in de paar jaar ergens aan mee, die € 5,- kan toch bijna iedere informatica-alumnus wel missen? Zo houd je toch nog wat binding met je wetenschappelijke roots en af en toe contact met vrienden uit je studietijd.

Johan Noltes
voorzitter@eniac.utwente.nl

November was een mooie maand voor alumni met vele activiteiten. Op 12 november vond het ENIAC afstudeerdersevent plaats, gecombineerd met de uitreiking van de ENIAC Scriptieprijs 2011. Op 26 november sloot de Universiteit Twente haar lustrumjaar af met een alumnidag voor alle oud-studenten van onze universiteit.

Wat mij betreft was het ook de maand van de 'communities'. Het was opvallend hoe groepen mensen elkaar weten te vinden om ervaringen uit te wisselen en bij te praten. Als student is dat een redelijk vanzelfsprekend proces, je ziet je medestudenten regelmatig op de universiteit, en hebt daar ook alle ruimte om elkaar te spreken. Maar als alumnus verandert dat toch echt. In een nieuwe omgeving bouw je ook weer een nieuwe community op. Het is dan de kunst om ook contact te houden met je oude communities.

Een mooi voorbeeld van het contact houden met oude communities was de alumnidag. Tijdens de reünie die in het teken stond van het UT50-lustrum kwamen ook veel verschilde communities bij elkaar. Er werden ervaringen uitgewisseld, maar ook nieuwe contacten opgedaan. En het is natuurlijk een mooie gelegenheid om terug te kijken op je fantastische studieperiode. Want na je afstuderen veranderd je leven natuurlijk behoorlijk.

Ik merk die verschillen zelf ook, nu ik net klaar ben met studeren en begonnen ben met werken. Je houdt je opeens bezig hele andere zaken. Waar we ons vroeger op verjaardagen druk maakten over welke kroeg we daarna gingen bezoeken, praten we tegenwoordig op de volgende housewarming over de bijtelling van onze leaseauto's. Alhoewel dat voor sommigen nog nooit helemaal zal wennen.

Als bestuur van ENIAC proberen we het bij elkaar brengen van deze communities te faciliteren. Daar zijn we mee begonnen tijdens het afstudeerdersevent; om afstuderende studenten tips en ervaringen uit te laten wisselen. Dat bleek een groot succes, de aanwezigen gingen volop in discussie over hoe je dat afstudeerproces nu het beste aan kunt pakken. En eerlijkgezegd had ik die tips ook best graag willen horen voordat ik ging afstuderen.

Maar we hebben als bestuur ook veel geleerd van deze evenementen. Die ervaring willen we graag meenemen in het nieuwe jaar, 2012 staat alweer voor de deur. Tijd voor een nieuw jaar, nieuwe kansen en een nieuw beleidsplan. Daar willen we graag jouw input bij gebruiken. Laat ons gerust weten wat je van ons verwacht, en hoe we jou met je oude community in contact kunnen brengen. Dan kunnen we op de ALV van 25 februari 2012 een beleidsplan presenteren waarin ook jouw visie vertegenwoordigd is. Op weg naar 2012, maak er een mooi jaar van!

Johan Noltes, voorzitter

Tutorial Node.js


Bas Stottelaar
Redacteur I/O Vivat

NODE.JS,JAVASCRIPT,IRC,PANDO-RABOTS

JavaScript, maar dan server-side

Wie dacht dat JavaScript alleen in een browser leefde, heeft het mis. Hoewel 99% van het JavaScript-gebruik in een browser is om pagina's dynamisch te maken, is er sinds de introductie van de opensource Google V8 JavaScript Engine ook een project genaamd Node.js. Dit project heeft als doelstelling om de taal ook los van een browser te kunnen gebruiken, voornamelijk 'server side'. In deze tutorial wordt getoond hoe een eenvoudige IRC-client te programmeren is met behulp van Node.js. We gaan er vanuit dat de lezer enigszins bekend is met het gebruik van JavaScript (en/of jQuery) en een lichte interesse in IRC heeft.

Installatie

Node is een commandline interpreter. Het heeft daarom een eenvoudig programma nodig om programmacode uit te voeren. Op het moment van schrijven is versie v0.4.12 te verkrijgen onder de Unix-platformen. Voor een onstabiele v0.5.9 is ook een build voor Windows beschikbaar. Zie hiervoor http://www.nodejs.org. Onder Unix: raadpleeg je favoriete packet manager zoals apt-get of homebrew om Node.js eenvoudig te installeren.

Om de installatie te testen open je een terminal en typ: '`node -v`'. De uitvoer moet het versienummer zijn van de geïnstalleerde versie van Node.js. Via de interactieve console (door enkel 'node' te starten) kun je experimenteren met de commando's. Deze console sluit je af met CTRL + D.

Hello world

Er is geen enkele programmataal dat geen 'Hello world' kent als de eerste stapjes. Ook in deze tutorial beginnen we met zo'n voorbeeld. Open in je favoriete tekstbewerker een nieuw bestand en neem de volgende inhoud over:

```
console.log("Hello world");
```

Sla het bestand op onder de naam '`run.js`'. Open vervolgens weer een terminal en start je programma met '`node run.js`'. De uitvoer van het programma moet '`Hello world`' zijn.

Node.js onderscheidt zich van andere interpreters

We gaan het programma uitbreiden. Node.js onderscheid zich van andere interpreters doordat het event-based is. Dit betekend dat aan elke reactie een gebeurtenis hangt die uitgevoerd kan worden. Een hele eenvoudige is het gebruik van een teller. Modificeer bovenstaande programma naar de code hiernaast.

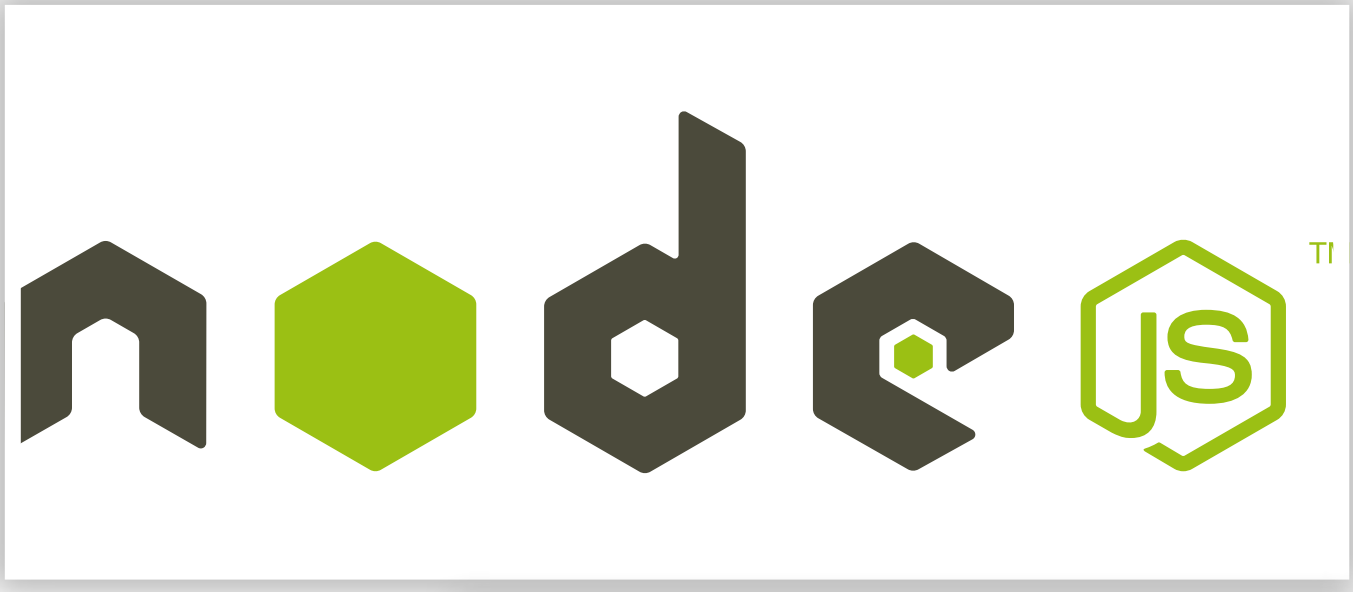
De uitvoer van dit programma is een oneindige teller. Zoals bij veel talen definieer je ergens een ingangspunt (main-methode). Node.js doet hier niet aan en begint bovenaan. De functie `setInterval(callback, timeout)` heeft twee parameters. De callback is een gebeurtenis die uitgevoerd wordt wanneer de timeout (in milliseconden) voorbij zijn. We kunnen ons voorstellen dat, in geval van een complexer voorbeeld, de gebeurtenis weer andere gebeurtenissen kan uitlokken, die op hun beurt weer hetzelfde doen. Uiteindelijk ontstaat er dan een hele traptrede aan gebeurtenissen waardoor structuur erg lastig te hanteren is. Het is daarom mogelijk om de functie (die nu geen naam heeft), buiten de `setInterval`-methode te halen, te voorzien van een naam (bijvoorbeeld foo) en de code aan te passen naar het volgende:

```
setInterval(foo, 1000);
```

Dit doet exact hetzelfde. Van belang is dat '`foo`' wel gedefinieerd is vóór de methode `setInterval`. Node.js werkt namelijk van boven naar beneden en volgorde is van belang. Nog een belangrijk voordeel van functies naar buiten halen, is het voorkomen van 'dubbele code'. De callbackfunctie kan op meerdere plekken (her)gebruikt worden.

```
var i = 0;

setInterval(
    function() {
        i ++;
        console.log(i);
    }
    , 1000
);
```



Prototype

JavaScript (en dus ook Node.js) is een prototype-gebaseerde taal in plaats van de class-gebaseerde taal. Dit betekend dat JavaScript geen klassen en constructors kent. Toch kan het wel een soort van class-gebaseerde manier van programmeren vertonen door functies. Hierbij komt de prototype-operatie ook van pas. Onderstaand voorbeeld laat zien hoe dit in z'n werk gaat:

```
function Auto() {
    this.kleur = "Rood";
}

var bmw = new Auto();
console.log(auto.kleur); // Rood

bmw.prijs = 1000; // Gaat mis, prijs bestaat niet
Auto.prototype.prijs = 500;
console.log(bmw.prijs); // 500

Auto.prototype.rij = function(){ .. }

var audi = new Auto();
bmw.rij();
audi.rij();
```

Kort samengevat: het prototype-object staat het toe nieuwe variabelen EN functies toe te voegen aan bestaande objecten (functies e.d.). Wat heeft dit voor een voordeel? Op deze manier is het bijvoorbeeld mogelijk om ingebouwde objecten zoals String en Array uit te breiden met eigen functies die voor alle instanties hiervan gelden. Het is dus heel eenvoudig om functionaliteit te 'injecteren'. Het is ook mogelijk om objecten ex-nihilo aan te maken. Dit gebeurd op een iets andere manier, zoals weergegeven bovenaan de pagina.

```
var obj = { rij : function() { .. }, prijs : 1000,
kleur : "Blauw" }
obj.rij();
console.log(obj.prijs); // 1000
```

Nog even een noot over het gebruik van 'this'. In elke functie refereert 'this' naar de scope van de functie van het huidige niveau. Ga je dus een functie in een functie plaatsen, dan verwijst 'this' in de inner-functie NIET naar de 'this' van de outer-functie. Dit kan wel als volgt opgelost worden:

```
function a() {
    var self = this;
    this.kleur = "Blauw";

      function b() {
        console.log(this.kleur); // Undefined - buiten
outer scope
        console.log(self.kleur); // Blauw
      }
}
```

Dit is van belang (en handig) in het geval van callbacks. Stel, functie a wordt aangeroepen met parameter y. Vervolgens dient b als een gebeurtenis die aangeroepen wordt wanneer (bijvoorbeeld) verbinding gemaakt is met een server. Dan kan er in de functie b nog steeds verwezen worden naar parameter y van functie a. Hetzelfde geldt voor de variabelen die in a gedefinieerd worden, zoals 'self'.

IRC

Met bovenstaande 'voorkennis' is het mogelijk om aan het echte werk te beginnen. De voorkennis is in wezen de basis van een gemiddeld Node.js-programma. We maken een eenvoudige IRC-client met de functionaliteit van een (echte chat)bot. Niet alle code zal besproken worden, maar wel de essentiële onderdelen. De volledige broncode is te downloaden vanaf https://github.com/basilfx/nodejs-ircbot.

```
Client.prototype.connect = function() {
    var self = this;

    self.socket.connect(this.config.port, this.config.server);

    self.socket.on("connect", function() {
        self.log("--", "Connected to " + self.config.server);

        self.send("NICK " + self.config.nick);
        self.send("USER " + self.config.user + " 8 * :" + self.config.real);

        self.onConnect();
    });

    self.socket.on("data", function (data) { self.receive(data) });
}
```

De implementatie bestaat uit drie onderdelen: IRC-client, Pandorabots-client en het programma zelf. Hoewel de code werkt, is er in het project geen foutafhandeling aanwezig. We beginnen met de IRC-client.

```
var net = require("net");
var irc = exports;

var Client = irc.Client = function(config) {
    var socket = new net.Socket();

    socket.setEncoding("ascii");
    socket.setNoDelay();

    this.config = config;
    this.socket = socket;
    this.buffer = "";
}
```

Maak een nieuw bestand genaamd 'irc.js' en sla deze alvast op. Node.js bevat een groot aantal ingebouwde modules, waaronder voor TCP/IP-communicatie. We beginnen dus ook om deze module kenbaar te maken in ons project en aan een soort van namespace te binden, namelijk 'net' (regel 1). Alle methodes in deze module kunnen nu benadert worden via de variabele 'net'. De regel daaronder geeft aan dat alles wat je onder de variabele 'irc' declareert, naar buiten (dus in een andere module) geëxporteerd wordt, via 'exports'.

Misschien dat het een wat rare declaratie lijkt op regel 4, maar dit is een manier om aan te geven dat 'Client' hetzelfde is als 'irc.Client' en dat hier een functie aan gekoppeld wordt. Het is meer een handigheidje dan een verplicht iets. Alles onder 'Client' is ook prima te benaderen via 'irc.Client'. In de functie zelf maken we een nieuwe socket en zetten we 'No Delay' uit, zodat alle data dat we verzenden niet eerst gebufferd wordt.

Nieuwe functies worden dus aan de 'prototype' van 'Client' gehangen. Bovenaan de pagina staat de methode om de connectie uit te voeren. Je ziet op regel 12 dat er pas echt verbinding gemaakt wordt. Regel 2 is een praktijkvoorbeeld over hetgeen wat onderaan de vorige pagina uitgelegd is. Het is namelijk zo dat 6 – 13 een interne functie is, en de variabele this dus niet naar de outer functie verwijst. Toch willen we die kunnen benaderen.

Regel 12 roept de functie 'onConnect' aan. Hoewel Node.js het prefereert om events aan te roepen via de module 'Events', is er voor de ouderwetse manier gekozen. In de broncode staat nu namelijk (helemaal onderaan) hoe de functie 'onConnect' gedefinieerd is, waardoor het een kwestie van kopiëren en plakken is om deze te overriden.

Iemand zou kunnen vragen of regel 15 ook anders geschreven zou kunnen worden, zoals onder deze alinea. Hoe-

```
Client.prototype.receive = function(data) {
    this.buffer = this.buffer + data;

      while (this.buffer) {
          var offset = this.buffer.indexOf("\n");
          if (offset < 0) return;

         var message = this.buffer.substr(0, offset);
            this.buffer = this.buffer.substr(offset + 2);
            this.log("<<", message);

            var match = message.match(/(?:(:[^\s]+) )?([^\s]+) (.+)/);
            if (match) this.handleData(match);
      }
}
```

```
Client.prototype.handleData = function(match) {
    var parameters = match[3].match(/(.*?) ?:(.*)/) || null;
    var info = match[0].match(/^:(.+)!~(.+)@(.+)/) || null;

    switch (match[2]) {
       case "JOIN":
          // Determine if we have joined, or someone else
          if (info[1] == this.config.nick)
             this.onJoin(parameters[2]);
          else
             this.onUserJoin(parameters[2], info[1]);

       break;
    }
}
```

wel dit valide, is het probleem dat in de functie 'receive', een verwijzing naar 'Client' verwacht wordt, maar deze niet meer bestaat omdat 'receive' aangeroepen wordt vanuit een andere context, namelijk de module 'net'.

```
self.socket.on("data",
self.receive);
```

Elke keer dat er data ontvangen wordt, wordt deze door de functie onderaan de vorige pagina verwerkt. Een respons van de IRC-server eindigt altijd op een newline-karakter. We kunnen data dus daar op splitsen en elke regel die daar niet op eindigt opslaan in de buffer. Vervolgens verwerken we de data regel voor regel.

De data die retour komt kan er ongeveer als volgt uitzien:

```
:Vivatbot!~Vivatbot@
localhost JOIN :#vivat

PING :irc.basilfx.net
```

Algemeen is dit te interpreteren als een optionele gebruikersnaam vooraf, gevolgd door een commando (kan ook numeriek zijn) en vervolgens extra parameters. Hierbij geeft een : aan dat data spaties kan bevatten. Door middel van een regular expression op regel 12 wordt deze data gesplitst waardoor het gemakkelijk is om het commando uit te lezen. Elke match, als dit er is, wordt doorgegeven aan de functie 'handleData'.

De methode handleData is een vrij lange functie. Hier wordt bepaald welke commando's de server wel of niet ondersteund. We beperken ons in dit geval tot een enkel voorbeeld, namelijk het commando 'JOIN' dat we van de server terugkrijgen als de client zelf (of een ander) een kanaal binnenkomt.

Zoals te zien is op regel 5, bevat de variabele 'match[2]' het commando. Op basis hiervan wordt bepaald wat er gedaan gaat worden. Omdat de server hetzelfde commando teruggeeft voor het joinen van onszelf als voor een ander, vergelijken we de nickname die meegestuurd wordt. Als deze gelijk is aan onze nickname, dan weten we natuurlijk dat wij het kanaal binnenkomen. Ook hier wordt weer een gebeurtenis aangeroepen, namelijk de gebeurtenis 'onJoin' met als parameter het kanaal waarin.

Hoe we een kanaal betreden, of een bericht versturen naar de server, laten we achterwege. Dit zijn namelijk eenvoudige functies die geen uitleg vereisen.

Pandorabots

Op het internet is de website http://www.pandorabots.com te vinden als een stal voor verschillende chatbots. Door middel van een API is hier gemakkelijk mee te communiceren. We gebruiken onze IRC-client als een interface voor deze chatbots.

Heel veel uitleg is er bij deze functie niet nodig. De exacte code is te vinden in 'chat.js'. Om met de website te communiceren moet er een URL aangeroepen worden in de onderstaande vorm.

```
Bot.prototype.say = function(input, callback) {
    var self = this;
    var data = querystring.stringify({
        botid : this.botId,
        custid : this.customerId,
        input: input
    });

    var handleResponse = function(response) {
        if (response.statusCode != 200) return;

        response.on('data', function(data) {
            temp = data + "";
            match = temp.match(/<that>(.+)<\/that>/);
            if (match) callback(match[1]);
        });
    }

    var options = {
        host: "www.pandorabots.com",
        port: 80,
        path: "/pandora/talk-xml?" + data,
        method: "GET",
    }

    http.get(options, handleResponse);
}
```

Het respons is een XML-document.

```
http://www.pandorabots.com/pandora/talk-xml?botid=[botId]&custid=[customerId]&input=[input]
```

Op regel 3-7 bouwen we de parameters voor deze URL op. Regel 8-15 is een callback voor de te ontvangen data. Node.js bevat standaard geen XML/DOM-module, waardoor we met een eenvoudige regular expression (regel 12) de respons parsen. Deze respons wordt teruggegeven aan de callback. Op regel 22 wordt alles in gang gezet. Dit is een eenvoudige functie uit de module 'http'. Hier wordt een http-request gedaan naar de server met een callback naar regel 8-15.

Het laatste stukje code is de code in het bestand 'run.js'. Deze hangt alles samen om er een geheel te maken.

```
var config = require("./config.js").config;
var irc = require("./irc.js");
var chat = require("./chat.js");

var irc = new irc.Client(config);
var bot = new chat.Bot("d689f7b8de347251");

irc.onWelcome = function() {
   irc.join("#vivat");
}

irc.onMessage = function(channel, nick, message) {
    if (channel.substr(0, 1) != "#") return;

    if (message.substr(0, 1) == "!") { // Command
        // ...
    } else { // Conversation
        bot.say(message, function(response) {
            irc.message(channel, response);
        });
    }
}

irc.connect();
```

Regel 1-3 initialiseert alle modules. De inhoud van die modules is via de functie 'exports' kenbaar gemaakt. Regel 4-5 maakt een nieuwe instantie aan van een IRC-client en de bot. De parameter bij 'bot' is een ID van een Pandorabot (zie http://www.pandorabots.com/botmaster/en/mostactive). De gebeurtenis 'onWelcome' wordt aangeroepen wanneer we door de IRC-server welkom worden geheten. Dit is niet het meest spannende. Het echte werk gebeurd in 'onMessage'.

Zie de originele code voor de volledige implementatie. Regel 12 controleert of we in een kanaal een gesprek voeren, of een privé-gesprek. Het gaat om de regels 18-20, waarin we de Pandorabot benaderen om aan de hand van het gezegde een respons genereren en deze terugsturen.

De allerlaatste regel zet alles in werking. Start het programma via 'node run.js'. Happy chatting!

Conclusie

Node.JS heeft laten zien dat het een volwassen taal is om snel efficiënte serverapplicaties te ontwikkelen door middel van het gebeurtenissenmodel. Het doel van Node.js is niet om een programmeertaal te zijn waarbij grafische applicaties ontwikkeld worden, maar juist om eenvoudig client/server-applicaties op te zetten. Met een eenvoudige IRC-client hebben we dit laten zien, waarin verschillende JavaScript technieken zoals prototype en ex-nihilo een belangrijke rol spelen. Van Node.JS verwachten we aankomende jaren nog zeker meer! Er zijn steeds meer uitbreidingen (modules) beschikbaar, waardoor er voor ieder wat wils is. Mocht je voortaan een simpele server nodig hebben, kijk dan ook eens naar Node.js!

Bronnen

Node.js v0.4.12 Manual & Documentation
http://nodejs.org/docs/v0.4.12/api/

RFC 2812 - Internet Relay Chat: Client Protocol (2000)
http://www.faqs.org/rfcs/rfc2812.html

Learning Server-Side JavaScript with Node.js (2010)
http://net.tutsplus.com/tutorials/javascript-ajax/learning-serverside-javascript-with-node-js/

Op bezoek bij Avanade

Erwin
de Moel
Solution Developer

Erwin de Moel is Solution Developer bij Avanade, een joint venture tussen Accenture en Microsoft. Kijk voor meer informatie op www.avanade.nl.

"Werken is eigenlijk gewoon betaald studeren…"

Kan je iets meer vertellen over wie je bent?

Mijn naam is Erwin de Moel en ik ben 27 jaar. In 2001 ben ik begonnen aan mijn Hbo opleiding "Information Engineering" aan de Hogeschool van Utrecht (vestiging in Amersfoort), een studie die veel lijkt op bedrijfsinformatietechnologie. Na deze opleiding had ik de keuze om te gaan werken of om verder te gaan met studeren aan de universiteit. Aangezien ik het gevoel had dat ik in mijn Hbo studie een stuk technische diepgang mistte, heb ik besloten om de master "Computer Science" aan de Universiteit Twente te gaan volgen.

Ik ben in september 2005 begonnen aan mijn masteropleiding. Tijdens deze studie ik onder andere in het bestuur van v.v. Drienerlo en de Centrale BewonersRaad. Daarnaast heb ik een aantal evenementen georganiseerd, waaronder de "KamerZoekDagen" 2006 en 2007, en ook het internationale zaalvoetbaltoernooi TISC 2005 (Trommel Indoor Soccer Cup).

Wat vond je van je studie?

De voorstelling die ik van de studie had, bleek uiteindelijk iets anders dan de werkelijkheid. Ik wilde graag meer weten over de technieken in de Software Development, maar op de universiteit ging men vooral in op de theoretische achtergrond van Computer Science. Ondanks dat de studie niet in lijn was met mijn verwachtingen, heb ik wel een andere belangrijke vaardigheid geleerd, namelijk het leren om een ingewikkeld en onbekend onderwerp eigen te maken. Dat is naar mijn idee het grote verschil tussen Hbo en universiteit. Bij het Hbo leerde ik het gebruiken van tools om zaken toe te passen, terwijl je bij de universiteit "leert om te leren".

Mijn ervaring is dat elk vak bestaat uit een leercurve, waarbij je de eerste paar weken geen idee hebt waar het over gaat, maar naarmate je meer met de stof gaat stoeien het begrip uiteindelijk exponentieel toeneemt. Het leuke is dat je in het bedrijfsleven constant met dezelfde leercurve te maken hebt. De eerste drie weken bij een nieuwe klant begrijp je vrij weinig van het nieuwe onderwerp, maar aangezien je het trucje al zo vaak hebt toegepast tijdens je studie heb je geleerd om hiermee om te gaan en blijf je vol vertrouwen doorgaan. Je leert omgaan met nieuwe problemen aanpakken, in plaats van standaard antwoorden voor problemen te leren. Zo ben je veel breder inzetbaar en kun je dus praktisch bij elke klant en onder elke omstandigheid aan de slag.

Wanneer je je thuis gaat voelen in het onderwerp, weet je dat het tijd is om op te stappen en een nieuwe uitdaging op te zoeken. Dit gebeurt in de vorm van een nieuw project, vaak bij een andere klant en mogelijk in een heel andere bedrijfssector. Op die manier blijf je nieuwe dingen aanleren, ontwikkel je jezelf maximaal en blijft het werk wat wij doen uitdagend. Voor mij is werken eigenlijk gewoon betaald studeren.

Hoe ben je na je studie verder gegaan?

Eigenlijk had ik na mijn studie weinig zin om te gaan werken. Naast mijn studie had ik een hobby online pokeren ontwikkeld. Dit groeide op een gegeven moment uit tot een degelijke bijbaan, waar ik een leuk salaris aan overhield. Toen ik afgestudeerd was, ben ik een aantal maanden fulltime gaan pokeren, waar ik mijn leven prima van kon betalen.

Op een gegeven moment kreeg ik via een oude studievriend via LinkedIn een berichtje dat hij bij een leuk bedrijf werkt, namelijk Avanade. Hij was daar 1,5 jaar geleden begonnen, na zijn studie aan de UT, en hij wist dat ik een affiniteit had met Microsoft technologie. Volgens hem was Avanade om die reden het perfecte bedrijf voor mij!

Enigszins sceptisch heb ik toegezegd om een open gesprek te houden met de manager "Solution Development", Eric Hol. Aangezien ik niet echt op zoek was naar een baan, stond ik er vrij open-minded in. Wel had ik besloten dat als ik bij Avanade zou gaan kijken, ik ook een beeld moest hebben van de rest van de arbeidsmarkt. Daarom heb ik 2 dagen mijn CV op Monsterboard gezet. Dankzij de hoge vraag naar ICT professionals, werd ik overspoeld door recruiters en bedrijven die graag een sollicitatiegesprek wilden hebben. Binnen no-time had ik dus 8 gesprekken gepland staan.

Uiteindelijk bleek dat er geen enkel bedrijf was die de uitdaging en ontwikkelmogelijkheden bood die Avanade wel te bieden had. Dat in combinatie met een heel open en goede samenwerkingssfeer en de gedrevenheid om resultaten te behalen, hebben mij doen besluiten om bij Avanade te komen werken.

Wat zie je als het grootste verschil

tussen werken vroeger en nu?

Een belangrijk verschil is de mate waarin medewerkers gestuurd worden. Vroeger stuurden bedrijven aan op aanwezigheid, in de vorm van bijvoorbeeld een prikklok waarmee medewerkers hun aanwezigheid konden aantonen. Tegenwoordig stuurt men aan op resultaten. Het maakt niet meer uit waar of wanneer je werkt en hoeveel uren je precies besteedt, "as long as you get the job done". Tijd is simpelweg een middel geworden om deze resultaten te behalen.

Bij Avanade ben je zelf verantwoordelijk voor je eigen agenda. Ik bepaal zelf wanneer ik werk en waar. Of ik nu bij de klant, op kantoor, of thuis zit, maakt voor mijn "baas" niet uit. Sterker nog, ik heb helemaal geen "baas". Vroeger hadden we managers die ons gingen vertellen wat we moesten doen. Tegenwoordig ben je je eigen baas en bepaal je zelf wat jij denkt dat op een moment het belangrijkste is. Uiteraard gaat dit in overleg met mensen die de projecten leiden waar je op dat moment aan werkt.

Ik heb ooit een artikel gelezen, waarin men onderzocht heeft wat mensen motiveert. Hieruit bleek dat drie factoren leiden tot betere prestaties en persoonlijke voldoening, namelijk Purpose, Autonomy en Mastery. "Purpose" is het gevoel dat we graag hebben om samen te werken aan een algeheel belang. Neem bijvoorbeeld Skype. Hun bedrijfsmotto is: "Our goal is to be disruptive, but in the cause of making the world a better place". "Autonomy" betekent de wil om zelfsturend te zijn en de controle over ons eigen leven te hebben. "Mastery" is de wil om beter te worden in bepaalde vaardigheden. Dat is de reden waarom mensen graag in hun vrije tijd muziekinstrumenten leren bespelen. Het is leuk om te zien dat je ergens beter in wordt, wat voldoening geeft.

Hoe zitten die trainingen in elkaar?

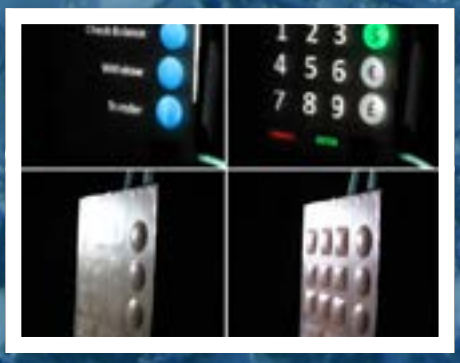
Bij Avanade krijgt iedereen jaarlijks 15 trainingsdagen, die hij/zij mag besteden aan trainingen die jou verder helpen in je ontwikkeling. Samen met je "Career Manager", een persoon die jou begeleidt bij je carrière, stel je elk jaar een trainingsplan op die je 2 keer per jaar evalueert. Aan het eind van elk jaar wordt vervolgens gekeken hoe in dat jaar jouw prestaties zijn geweest, op basis van project reviews, feedback van klanten en collega's en het behalen van je trainingsdoelen. Deze prestaties worden gemeten aan alle andere mensen op jouw level, om op die manier de beste medewerkers te laten promoveren naar het volgende level.

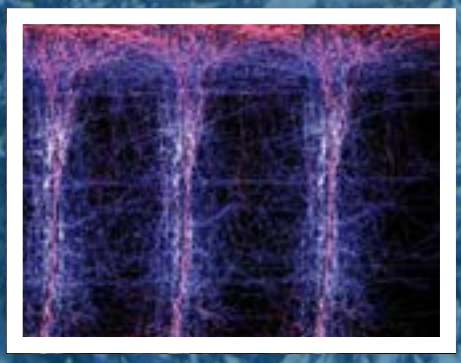
Daarnaast heb je zelf invloed op nieuwe projecten die je gaat doen. Als je een bepaalde interesse hebt, dan kun je daarin trainingen volgen en aangeven bij de afdeling "Scheduling" dat je jezelf graag in een bepaald gebied verder wilt ontwikkelen. Daar wordt dan rekening mee gehouden bij het inplannen van een vervolgproject. Op een gegeven moment bouw je een netwerk op van projectmanagers op die vertrouwen in jou hebben en regel je via deze weg je eigen projecten. Daarmee ben je niet langer afhankelijk van de projecten die de afdeling "Scheduling" voor jou regelt en heb je zelf controle over je carrière; "direct thy self".

Naast al deze ontwikkelmogelijkheden en vrijheden, vind ik het leukste aan werken dat men de skills en vaardigheden van verschillende mensen combineert, om samen tot een beter resultaat te komen. Zo is de ene persoon bijvoorbeeld ontzettend goed in het technisch ontwikkelen van systemen, terwijl een ander veel beter is in communiceren met klanten en business requirements in kaart brengen. Je merkt een correlatie tussen wat men leuk vindt en waar men goed in is. Op die manier zie je dat iedereen binnen een project op zijn eigen manier een bijzondere bijdrage levert aan het geheel. Werken bij Avanade heeft mij in ieder geval één heel belangrijk ding geleerd, namelijk: "The whole is greater than the sum of its parts".

VOLGENDE KEER IN I/O VIVAT

- **PNEUMATIC DISPLAYS**
- **HYPOTHES.IS**
- **BRAIN ON A CHIP**





Advertentie
Topicus