Jan Vlietland van Nederlands Instituut voor de Software Industrie over de drie A's van applicatiebeveiliging

6 juli 2018

In het eerste deel van deze korte serie over het ontwikkelen en bouwen van veilige applicaties lichtte Jan Vlietland van het Nederlands Instituut voor de Software Industrie negen maatregelen toe om dit doel te bereiken. In het tweede deel gaan we in op de drie A's van applicatiebeveiliging: authenticatie, autorisatie en accountability.

"Over authenticatie, autorisatie en accountability valt heel veel te vertellen. Maar laten we voor de goede orde beginnen met vast te stellen wat we eigenlijk onder deze drie termen verstaan", zegt Jan Vlietland van het NISI.

Bovenstaand is de inleiding van het artikel zoals dit in Infosecurity Magazine (en CloudWorks) is verschenen. In het onderstaande wordt het vervolg van het artikel opgesplitst en voorzien van de oorspronkelijke tekst uit mijn PowerPoint-slides waaruit overduidelijk het plagiaat blijkt.

Tekst artikel auteur NISI:

- **Authenticatie** is het proces van het verifiëren van de identiteit van een gebruiker. Dit kan een persoon zijn, maar natuurlijk ook een – zeg maar – apparaat, een applicatie of een microservice. Dit gebeurt bij voorkeur in realtime.
- **Autorisatie** is de procedure die wordt gehanteerd om vast te stellen wat een bepaalde geautoriseerde gebruiker mag doen. Zeg maar: welke rechten heeft die gebruiker?
- **Accountability** geeft aan dat we willen vaststellen wie of wat verantwoordelijk is voor een actie en hoe we deze gebruiker daarvoor aansprakelijk kunnen houden.

Mijn oorspronkelijke PowerPoint-slide:

- **Authentication** is the process of verifying the identity
  - Proof of identity
  - Mostly a realtime process
- **Authorization** is a further step in the process
  - What is the subject with a certain identity allowed to do?
- **Accountability** is about being able to determine who or what is responsible for an action and can be held accountable

Tekst artikel auteur NISI:

Vormen van authenticatie

Er zijn meerdere manieren om het proces van authenticatie te realiseren. De eerste is single factor authentication. Hierbij wordt slechts één factor (zeg maar: hulpmiddel) gebruikt om de identiteit van een gebruiker vast te stellen. Veelal gaat het dan om een wachtwoord. Daarmee is meteen duidelijk dat single factor authentication in feite een zwakke beveiligingsaanpak is. Er zijn tal van methoden om hier omheen te werken. Passwords kunnen geraden worden. Ze kunnen via malware gestolen worden, via social engineering of sniffing afhandig worden gemaakt, noem maar op. “Dan kennen we multi-factor authentication, wat al beduidend veiliger is dan de single-variant”, aldus Vlietland. In dit geval dient de gebruiker zich via twee verschillende hulpmiddelen te identificeren. Denk aan een combinatie van een wachtwoord en een PIN, een smartcard of een token, een vingerafdruk of een irisscan.” Maar het is ook mogelijk om bijvoorbeeld gedragskenmerken toe te passen of locatie-specifieke info. Of denk aan een identificatie via het te gebruiken apparaat of een tijdstip.

Mijn oorspronkelijke PowerPoint-slides:

- **Single factor authentication** is based on a single factor to verify the identity of the subject
- Most of the times the authentication factor represents a password
- Passwords are vulnerable to a great number of threats
  - Malware
  - Guessing
  - Dumpster diving
  - Shoulder surfing
  - Social engineering
  - Brute force attack
  - Dictionary attack
  - Rainbow table attack
  - Sniffing

- **Multifactor authentication** is the process of verifying the identity at least **twice**
- Based on a combination of at least **two different** identity components
  - Something you know
    - Password, answer on secret question, PIN, passphrase, virtual keypad
  - Something you have
    - Token, smartcard, digital certificate, mobile device, grid, TAN sheet
  - Something you are (biometric treats)
    - Fingerprint, face pattern, iris pattern, voice, behaverial treats (swipe dynamics, signature dynamics, keypad dynamics)
  - Something about your context
    - Location, time, device type

Tekst artikel auteur NISI:

Continuous authentication

Een derde vorm is wat we wel noemen: continuous authentication. Hierbij wordt de identiteit van een gebruiker voortdurend vastgesteld. Na een initiële log-in procedure wordt aan de hand van biometrische factoren, contextinformatie of bijvoorbeeld gegevens over het gebruikte apparaat continu gekeken wie de gebruiker is. Denk bij contextuele gegevens bijvoorbeeld aan een locatie (via bijvoorbeeld GPS-data) of een tijdstip. “Bij toepassing van continuous authentication wordt het voor een cybercrimineel veel lastiger om een sessie over te nemen of om met een gestolen apparaat toegang tot een applicatie of tot data te verkrijgen.”

Mijn oorspronkelijke PowerPoint-slide:

- **Continuous authentication** is the ongoing process of verifying the identity
- Based on a continuous verification of several identity attributes **after** the initial login
  - Biometric treats
  - Context (GPS location, time)
  - Device (smartphone, tablet etc.)
- With continuous authentication security will be improved
  - Session hijacking becomes much more difficult
  - When a device is snatched from the hand the damage will be less severe

Tekst artikel auteur NISI:

OAuth

Dan kennen wij uiteraard nog open authentication (OAuth). Vlietland: “Dit is een specificatie die het mogelijk maakt om toegang tot resources te verkrijgen zonder dat hierbij het gebruikte wachtwoord of andere vertrouwelijke gegevens bekend worden gemaakt. In plaats daarvan wordt een token gebruikt die alle relevante gegevens bevat.” OAuth is veel makkelijker te implementeren dan Security Assertion Markup Language (SAML), vertelt Vlietland. Het is ook beter geschikt voor de mobiele wereld. Er wordt gewerkt met RESTful API’s. “Daarnaast geldt dat OAuth gebruikt kan worden voor zowel authenticatie als autorisatie.

In figuur 1 is kort weergegeven hoe dit in de praktijk werkt.

Resource owner (user)	Provides access to resources (such as information in a social network)
Resource server	Server that contains the resources (such as a social network)
Authorization server	Server that provides access on behalf of the resource server, based on a token that is sent to the client Usually the same as the resource server
Client	Application or app to which access to the resource is assigned
Access token	Unique string of characters, on the basis of which the client can access resources Contains access information, resource scope and period of validity
Refresh token	Token that can be used to apply for a new token

Figuur 1. Het gebruik van OAuth in de praktijk.”

(Bovenstaande figuur met bijbehorende tekst werd verwijderd nadat ik in een commentaar onderaan het online artikel had gewezen op schending van het auteursrecht en het intellectueel eigendomsrecht)

Mijn oorspronkelijke PowerPoint-slides:

- OAuth is a specification with which access to resources can be provided without revealing a password or other confidential information
- Instead, a token is used that represents all necessary information
- Much more easy to implement than SAML and thus more suitable for the mobile world
- Based on RESTful (REpresentational State Transfer) API’s
- Makes use of HTTP verbs such as GET, POST, PUT, DELETE
- Can be used for both authentication and authorization

Resource owner (user)	Provides access to resources (such as information in a social network)
Resource server	Server that contains the resources (such as a social network)
Authorization server	Server that provides access on behalf of the resource server, based on a token that is sent to the client Usually the same as the resource server
Client	Application or app to which access to the resource is assigned
Access token	Unique string of characters, on the basis of which the client can access resources Contains access information, resource scope and period of validity
Refresh token	Token that can be used to apply for a new token

Tekst artikel auteur NISI:

Autorisatie

Er bestaan meerdere autorisatiemethoden. Denk onder andere aan:

- Mandatory Access Control of MAC waarbij gewerkt wordt met security labels
- Discretionary Access Control (DAC) waarbij de toegang tot de relevante informatie beheerd wordt door de eigenaar van die info
- Role-based Access Control (RBAC), waarbij de toegang tot informatie geregeld is op basis van de rol die een gebruiker vervult
- Attribute-based Access Control (ABAC), waarbij een attribuut dan bijvoorbeeld een locatie of een tijdstip kan zijn

Mijn oorspronkelijke PowerPoint-slide:

During the years a number of authorization methods have been developed

- Mandatory access control (MAC)
  - Access to information is based on security labels
- Discretionary access control (DAC)
  - Access to information is controlled by the owner
- Role-based access control (RBAC)
  - Access to information is based on roles (e.g., physician, administrator)
- Attribute-based access control (ABAC)
  - Access to information is based on attributes (e.g., time, location)
- And several others, most of which are invented by vendors

Tekst artikel auteur NISI:

“Er zijn dus tal van methoden, met ieder hun voor- en nadelen. Belangrijker hier is echter om vast te stellen dat in de praktijk het gebruik van deze access control-methoden nog regelmatig mis gaat.” Bekende problemen zijn het niet goed toepassen van het fenomeen ‘scheiding van taken’ (segregation of duties). Ook gebeurt het regelmatig dat de ‘least privilege’ en ‘need to know’ (zie deel 1 van deze serie) niet goed zijn geïmplementeerd. “Soms zijn de modellen die gebruikt worden voor het definiëren van rollen te complex of worden gast- en test-accounts verkeerd gebruikt. Of wordt onvoldoende aandacht besteed aan de manier waarop ‘events’ behandeld, gelogd en onderzocht worden. En zo zijn er nog meer punten die mis kunnen gaan – een account database die bijvoorbeeld niet up-to-date is.”

Mijn oorspronkelijke PowerPoint-slide:

- Segregation of duties (SoD) insufficiently applied
- Least privilege insufficiently applied
- Need-to-know insufficiently applied
- Too complex role models
- Poor event handling/logging/log viewing
- Wrong use of guest and test accounts
- Authorization creep
- Pollution of the account database (not up-to-date)

Tekst artikel auteur NISI:

RBAC

Laten we nog even nader kijken naar de methode van RBAC ofwel role-based access control. “RBAC vormt in feite een reeks van autorisaties die is gebaseerd op enerzijds de organisatiestructuur van de gebruiker, zijn business processen en een aantal policies en regels. Daarmee heeft deze manier van werken het in zich om het hele proces rond autorisatie transparanter te maken. Dat is dus een belangrijk pluspunt”, meent Vlietland. “Daar staat echter wel tegenover dat het implementeren van RBAC soms erg complex kan zijn en – mede daardoor – relatief veel tijd kost.”

Mijn oorspronkelijke PowerPoint-slide:

Role-based Access Control (RBAC) is the discipline of assigning access rights through roles

An RBAC role can be defined as a collection of authorizations based on

- Organizational structure
- Business processes
- Policies and rules

RBAC should make authorizations more transparant

However, implementing RBAC is often very complex and time-consuming

Tekst artikel auteur NISI:

Duidelijke voordelen

Toch zijn de voordelen van role-based autorisaties evident, meent hij. “Met RBAC is het toekennen, veranderen of intrekken van autorisaties veel efficiënter geregeld dan bij andere methoden. Is het eenmaal goed geïmplementeerd, dan zijn de autorisaties veel transparanter. Dat geldt met name voor de functionarissen die deze autorisaties moeten begrijpen: business managers, auditors en eindgebruikers.” Bovendien kunnen de principes die ten grondslag liggen aan een goede autorisatie veel gemakkelijker worden afgedwongen, stelt Vlietland. “Dat zijn dus: need-to-know, least privilege en segregation of duties.”

Mijn oorspronkelijke PowerPoint-slide:

With RBAC, assigning (changing, withdrawing) authorizations becomes more efficient

When implemented well, authorizations will become more transparent to those who need to understand them

- Business managers
- Auditors
- End users

Authorization principles (need-to-know, least privilege, segregation of duties) can be enforced in a more effective way