перешкоджають таким процесам або призводять до неефективного використання засобів на їх розробку, впровадження і захист. До найбільш значущих серед них слід віднести:

фактичну самоізоляцію України від міжнародного інформаційного співтовариства зважаючи на невідповідність законодавства і стандартів нашої держави світовим вимогам;

відсутність сумісності між ІТС різних відомств і організацій України, що призводить до надмірності у зборі первинної інформації, подорожчання розробок і експлуатації таких систем;

відсутність централізованої державної структури, що регламентує інформаційні процеси у нашому суспільстві тощо.

Дані проблеми суттєво впливають на створення комплексної системи захисту інформаційного і кіберпросторів України від внутрішніх і зовнішніх злочин і загроз, а також на можливість інтеграції нашої держави у світову інформаційну спільноту.

Список літератури

1. Денис Фери. Секреты супер-хакера.- СПб.: Издательский Дом "Невский прспект,1977.

2. Анин Б.Ю. Защита компьютерной информации. – СПб.:БХВ-Петербург, 2000. – 384 с. : ил.

3. Стюарт Мак-Клар, Джоел Скембрей, Джордж Куртц. Секреты хакеров. Безопасность сетей - готовые решения, 3-е издание. : Пер.с англ. - М. : Издательский дом "Вильямс", 2002. - 736 с. : ил.

4. Левин М. Библия хакера 2. Книга 1 (Книга 2). - М.: Майор, 2003. - 640 с. (- 688 с.)

5. Alex WebKnacKer. Быстро и легко. Хакинг и антихакинг: защита и нападение. Учебное пособие. - М.: Лучшие книги, 2004. - 400 с.: ил.

6. Скляров Д. В. Искусство защиты и взлома информации. – СПб.: БХВ-Петербург, 2004. – 288 с: ил.

7. Крис Касперски. Техника и философия хакерских атак – записки мыщ'а. – М.: СОЛОН-Пресс, 2004. – 272 с. : ил

8. Семёнов Ю.А. Обзор некоторых видов сетевых атак. [Електронний ресурс]. – Режим доступу: http://citforum.ru/nets/semenov/6/intrusion.shtml

9. Алексей Койнаш. Взлом и защита компьютерной сети: этапы и инструменты. [Електронний ресурс]. – Режим доступу: http://www.vlasnasprava.info/ru/business_az/how_to_grow/protect.html?_m=publications&_t=rec&id=748

10. Методы взлома компьютерных систем. Електронний ресурс]. – Режим доступу:

11. Биячуев Т.А. / под ред. Л.Г.Осовецкого Безопасность корпоративных сетей. – СПб: СПб ГУ ИТМО, 2004. – 161 с.

12. Степашкин М.В. Оценка уровня защищенности компьютерных сетей на основе построения графа атак /И.В.Котенко, М.В. Степашкин // Труды международной научной школы "Моделирование и анализ безопасности и риска в сложных системах". – Спб., 2006. – С.150-154

Рецензент: РибальськийО.В.

Надійшла 22.09.2011

Мельник Н. Д., Кльок О.В., Паршуков С.С.

(Інститут спец. зв'язку та зах. інф.)

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ WMI ДЛЯ СТВОРЕННЯ АВТОМАТИЗОВАНОГО ОБЛІКУ НОСІЇВ ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ

Технологія Windows Management Instrumentation (WMI) дозволяє за допомогою об'єктної моделі операційної системи та відповідних сценаріїв створити систему обліку використання зовнішніх носіїв, а структура таких сценаріїв буде повністю прозора.

Для виконання задачі обліку зовнішніх носіїв треба виконати наступні кроки: створити за допомогою технології WMI постійний споживач подій, який дозволить

провести автоматизований облік всіх логічних носіїв, створити та записати проведений облік зовнішніх та логічних носіїв в спеціально створену базу даних.

Для створення постійного споживача подій, що буде проводити облік всіх зовнішніх дисків треба виконати таку послідовність дій. Створити фільтр, який буде реагувати на підключення зовнішніх носіїв. Створити споживач подій, який буде виконувати перевірку зовнішнього носія та проводити запис номеру логічного диска в відповідну базу даних. Створити об'єкт-зв'язку що буде пов'язувати відповідний фільтр та споживач подій. Робота такого споживача подій буде проводитися тільки при проведенні інвентаризації зовнішніх носіїв.

Створення фільтра, що реагує на підключення зовнішніх пристроїв має наступний вигляд, що представлений на рис.1

```
1. Option Explicit
2. Dim objLocator, objServer, objFilterClass, objFilter
3. Set objLocator = CreateObject("WbemScripting.SWbemLocator")
4. Set objServer = objLocator.ConnectServer("localhost","root\cimv2")
5. Set objFilterClass = objServer.get("__EventFilter")
6. Set objFilter = objFilterClass.SpawnInstance_()
7. objFilter.name="FilterLD"
8. objFilter.querylanguage="WQL"
9. objFilter.query= "Select * from __InstanceCreationEvent within 5 where " &_
                    "TargetInstance ISA 'win32_LogicalDisk'"
10. objFilter.Put_
```

Рис.1. Створення фільтра, що реагує на підключення зовнішніх дисків.

Алгоритм створення фільтра наступний: підключитися до простору імен в якому знаходяться всі об'єкти керування операційною системою на локальному робочому місті (рядок 3 та 4), створити посилання на об'єкт який може працювати з фільтрами (рядок 5), згідно з вимогами технології WMI заповнити відповідні властивості фільтру та зберегти фільтр (рядки 6-10). Головним елементом у фільтрі є створення запиту (рядок 9), який пояснює сценарію, що у випадку підключення зовнішнього носія треба реагувати на це підключення.

Реагує на підключення – споживач подій. Сценарій створення споживача подій представлено на рис. 2.

```
1. Set objConsumerClass = objServer.get("ActiveScriptEventConsumer")
2. Set objConsumer = objConsumerClass.SpawnInstance_()
3. objConsumer.name="ConsLD"
4. objConsumer.ScriptingEngine="VBscript"
5. objConsumer.ScriptFilename="C:\scripts\MonitoringLD.vbs"
6. objConsumer.Put_
```

Рис.2. Створення споживача, що запускає зовнішні сценарії.

Порядок створення споживачів подій наступний: створити посилання на об'єкт, який може запускати зовнішні сценарії (рядок 1), згідно з вимогами технології WMI заповнити відповідні властивості споживача та зберегти всі налаштування (рядки 2-6). У випадку підключення зовнішнього диску виконується сценарій (рядок 5) за вказаним ім'ям та місцем знаходження. В цьому сценарії знаходиться процедура інвентаризації зовнішніх та логічних дисків.

Для того щоб об'єднати фільтр та відповідний споживач подій створюється елемент зв'язки. Сценарій створення зв'язки представлено на рис. 3.

```
1. Set objEventFilter = objServer.get("__EventFilter.name=""FilterLD""")
2. Set objEventConsumer = _
   objServer.get("ActiveScriptEventConsumer.name=""ConsLD""")
3. Set objBindingClass = objServer.get("__FilterToConsumerBinding")
4. Set objBinding= objBindingClass.SpawnInstance_()
5. objBinding.consumer=objEventConsumer.Path_
6. objBinding.filter=objEventFilter.Path_
7. objBinding.Put_
```

Рис.3. Створення зв'язки між фільтром відповідним споживачем подій.

Порядок створення зв'язки наступний: створити посилання на фільтр та відповідний до нього споживач подій (рядок 1,2), створити посилання на об'єкт зв'язку який може з'єднати відповідний фільтр та споживач (рядок 3), згідно з вимогами технології WMI заповнити відповідні властивості об'єкта зв'язки та зберегти всі налаштування (рядки 4-7).

При кожному підключенні нового зовнішнього носія фільтр буде відслідковувати цю подію, а завдяки зв'язці споживач подій буде виконувати сценарій ім'я та шлях до якого вказано в рядку 5 рис. 2.

Вказаний сценарій може проводити інвентаризацію логічних та зовнішніх дисків.

Сценарій, що проводить інвентарізацію дисків має наступні кроки (рис. 4). Визначення номеру підключеного зовнішнього логічного диску. Зчитування номерів санкціонованих носіїв з бази даних, формування відповідного масиву за допомогою процедури (рис. 5). Порівняння номеру підключеного зовнішнього логічного диску з номерами санкціонованих дисків зі створеного масиву за допомогою відповідної процедури (рис. 6). Запис номеру логічного диску до бази даних санкціонованих зовнішніх носіїв за допомогою процедури (рис. 7).

```
1. Set objLocator = CreateObject("WbemScripting.SWbemLocator")
2. Set objServer  = objLocator.ConnectServer("192.168.10.22","root\cimv2")
3. Set objLDClass = objServer.ExecQuery ("select * from win32_logicaldisk")
4. For Each objLD In objLDClass
5.    If objLD.volumeserialnumber <> Empty Then
6.    dbRead ()
7.    dbCmp objLD.volumeSerialNumber
8.    End If
9. Next
```

Рис. 4. Визначення номеру підключеного логічного диску

Номер підключеного логічного диску визначається рядком 5. В рядках 6 та 7 проводиться формування масиву всіх зареєстрованих носіїв та порівняння елементів масиву з номером підключеного диску.

```
1. Sub dbRead ()
2.    Set rs = CreateObject("ADODB.RecordSet")
3.    strConnectString = "Driver={Mysql odbc 5.1 driver}; " &_
4.                "server=192.168.10.22;UID=user;password=password;" & _
5.                "database=computers;option=3"
6.    rs.ActiveConnection = strConnectString
```

```
7.    On Error Resume Next
8.    rs.Open "select * from logdisk;"
9.    While not rs.eof
10.     m(i)= rs(1)
11.     i=i+1
12.     rs.MoveNext
13.   Wend
14. End Sub
```

Рис. 5. Процедура формування масиву санкціонованих зовнішніх носіїв.

В рядках 2-8 проводиться підключення до бази даних, що містить всі зареєстровані носії. В рядках 9-11 проводиться формування масиву зареєстрованих носіїв, при умові, що номера логічних дисків знаходяться в другому стовбці таблиці бази даних «logdisk» – рядок 10. В рядку 11 визначається кількість елементів масиву – кількість зареєстрованих носіїв.

Процедура запису номера логічного диску до бази даних потребує перевірки записаної інформації для виключення повторного запису (рис 6).

```
1.  Sub dbCmp (per3)
2.    p1 = True
3.    If i =0 Then
4.      dbWrite (per3)
5.    Else
6.      While j < i And p1 = True
7.        If per3 = m(j) Then
8.          p1 = False
9.        ElseIf per3<>m(j) And j = i-1 Then
10.         dbWrite (per3)
11.         p1=False
12.       Else
13.         j=j+1
14.       End If
15.     Wend
16.   End If
17. End Sub
```

Рис. 6. Порівняння підключених носіїв з існуючою базою данних.

Якщо в базі даних немає записів про санкціоновані носії – рядок 3, то проводиться запис підключеного логічного диску – рядок 4. Якщо в базі даних є відповідні записи то перевіряється кожна з них на відповідність з номером підключеного логічного диску – рядки 6-15. При відсутності запису номера підключеного логічного диску в базі даних проводиться відповідний запис – рядок 10 за допомогою процедури запису санкціонованих носіїв рис. 7.

```
1.  Sub dbWrite (per1)
2.    Set cmd = CreateObject("ADODB.Command")
3.    strConnectString = "Driver={Mysql odbc 5.1 driver}; " &_
4.                       "server=192.168.10.22;UID=user;password=password;" & _
5.                       "database=computers;option=3"
6.    cmd.ActiveConnection = strConnectString
```

```
7.    cmd.CommandText = "insert into logdisk (NLogDisk) value ('"& per1 &"');"
8.    cmd.Execute
9.  End Sub
```

Рис. 7. Процедура запису санкціонованих носіїв до бази даних

id	pid	date	time
1	253463666	2011-06-17	09:28:36
2	2147483647	2011-06-17	08:28:36
3	153455	2011-06-17	07:28:36

Рис. 8. Занесення ідентифікаторів USB – носіїв до БД

На рисунку 8 представлений фрагмент БД до якої заносяться всі ново підключені зовнішні носії пам'яті. Поле 'id' – являється ключовим полем котре інкрементується при додавання нового носія, поле 'pid' – містить персональний ідентифікатор кожного ново підключеного пристрою , поле 'date' – містить дату проведення операції. поле 'time' – час проведення операції.

Висновки

Таким чином в результаті створення відповідних сценаріїв можна проводити облік будь-яких зовнішніх носіїв, а в подальшому контролювати їх використання в автоматизованих системах класу Windows. Такий підхід забезпечить відповідний рівень захищеності автоматизованої системи на предмет використання виключно санкціонованих як носіїв так і будь-яких чи-то периферійних чи-то інших функціональних пристроїв. В подальшому планується розглянути механізм контролю використання зовнішніх носіїв для Unix подібних операційних систем.

Список літератури

1. Шетка Петр Microsoft Windows Server 2003. Практическое руководство по настройке сети. — СПб.: Наука и Техника, 2006. — 608 с.: ил.

2. Попов А.В. «Коммандные файлы и сценарии Windows script host».-Спб.: БХВ-Петербург, 2002-320с

3. www.wikipedia.org

4. http://google.com.ua

5. Попов А.В. «Коммандные файлы и сценарии Windows script host».-Спб.: БХВ-Петербург, 2002-320с.

6. Шварц Б., Зайцев П., Ткаченко В. и др. MySQL. Оптимизация производительности Символ-Плюс, 2010

Рецензент
Надійшла 27.05.2011