МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ

В.Б. Крижанівський

МЕТОДИЧНІ ВКАЗІВКИ ДЛЯ САМОСТІЙНОЇ РОБОТИ

з курсу «Безпека програм і даних»

для студентів напряму підготовки 6.050103
«Програмна інженерія»

Житомир

2012

ЗМІСТ

Модуль №1

Business Desktop Deployment (BDD)

Рішення Microsoft для розгортання настільних бізнес-систем Business Desktop Deployment (BDD). За своєю суттю BDD - це набір методичних вказівок і правил, заснованих на концепції Microsoft Solution Framework для планування, побудови, тестування і розгортання робочих місць користувачів в рамках корпоративної мережевої інфраструктури.

Останнім часом багато компаній, які мають значний парк комп'ютерів, стикаються з проблемою оновлення системного і прикладного ПЗ. Особливо гостро стоїть питання про доцільність-ності переходу на нові версії операційних систем та офісних пакетів Microsoft. Адже крім фінансових і тимчасових витрат, існують ризики, пов'язані з несумісністю використовуваних додатків з новим базовим ПО, зі збоями в уже налагодженому механізмі обробки даних і т.д.

Компанія Microsoft випустила комплексне програмне рішення Business Desktop Deployment (BDD). За своєю суттю BDD - це набір методичних вказівок і правил, заснованих на концепції Microsoft Solution Framework для планування, побудови, тестування та розгортання ра-бочих місць користувачів в рамках корпоративної мережевої інфраструктури.

Рис. BDD розділяє процес розгортання настільних бізнес-систем на конкретні завдання

BDD пропонує цілісну систему управління розгортанням і оновленням робочих місць користувачів. Схематично ця система представлена на малюнку. Вона включає в себе комплекс документів, що описують типові задачі і процеси управління робочими місцями, керівництво по сумісності додатків, керівництва по усуненню несправностей інфраструктури, систему створення та підтримки еталонних образів, засоби створення основних і допоміжних пакетів додатків, засоби міграції користувачів, налаштування захисту робочих станцій , а також рекомендації з організації впровадження, забезпечення доступності та оновлення систем.

Якщо ви серйозно зацікавилися вирішенням проблем розгортання мережевої інфраструктури на основі ОС Windows 2003/XP в своїй організації, то рекомендуємо встановити і вивчити набір посібників та інструментальних засобів BDD від компанії Microsoft (доступний для безкоштовного завантаження з сайту Microsoft).

Модуль №2

Сучасні напрямки розвитку технології RAID

Для початку трохи теорії, історії і термінології: RAID - Redundand Array of Inexpensive (зараз частіше вживають Independent) Disks, що буквально перекладається як "надлишковий масив з недорогих дисків" або "надлишковий масив з незалежних дисків". Це технологія, що дозволяє декілька реальних дисків представити як один віртуальний. Розмір результуючого диска залежить від способу комбінації цих дисків, так званого рівня масиву. Також, коли приступаєте до побудови RAID'у, не забудьте, що його можна будувати лише з одинакових за розміром дисків (є один вийняток, але він "приклеєний" до технології RAID чисто з технічних причин і, по суті, зі всіма решта рівнями має надзвичайно мало схожості). Ну і в залежності від обраного рівня Вам буде потрібно 2,3 або більше дисків.

Найросповсюдженішими є наступні рівні:

JBOD(Just a Bunch Of Disks) або Linear - власне і є отой вийняток, який дозволяє включати в нього будь-які диски. Масив будується звичайним лінійним способом: спочатку запис іде на перший диск, коли він заповнюється - на другий, і так далі. Жодного виграшу по надійності збереження даних чи швидкості читання/запису не забезпечує. Єдиний ефект - диск по розміру рівний сумі всіх учасників масиву.

RAID-0 - строго кажучи теж не точно відповідає абревіатурі RAID, бо не забезпечує жодної надлишковості.Може містити довільну кількість дисків більше двох. Дані записуються блоками по черзі на кожен диск, таким чином можна одночасно проводити стільки операцій читання/запису скільки є дисків, відповідно, у стільки ж разів зростає сумарна швидкість цих операції. Як і в JBOD розмір дорівнює сумі розмірів учасникі, надлишковості теж немає, зате є приріст у швидкості.

RAID-1 - також відомий як Mirror - дзеркало. Кожен блок даних записується на всі диски. Теж може містити довільну кількість дисків (два і більше). Зберігає інформацію при виході з ладу всіх дисків крім одного. Розмір масива дорівнює розміру одного диска-учасника, швидкість запису відповідає швидкості запису на найповільніший з дисків-учасників, зате швидкість читання може бути така ж як і в RAID-0 а деколи і більша (хоча особливо ліниві виробники цей ефект не використовують), це досягається за рахунок того, що різні блоки, як і в RAID-0 можна одночасно зчитувати з різних дисків, але якщо у RAID-0 цей фокус проходив лише при лінійному читанні, то в RAID-1 це працює при будь-яких умовах, адже на кожному диску є будь-який потрібний блок.

RAID-1+0 або **RAID-10** - також частково вийняток з класифікації, бо по-факту є комбінацією RAID-1 і RAID-0: декілька масивів RAID-1 об’єднуються в RAID-0. Зазвичай частинки RAID-1 у ньому робляться з двох дисків (при автоматичній конфігурації - завжди), тому прийнято вважати, що він потребує парну кількість дисків не менше чотирьох. Обчислення розміру і швидкості відповідно слід виконати для всієї послідовності побудови. В класичному мінімальному варіанті з 4-х дисків: розмір як у двох дисків, швидкість запису вдвічі більша ніж на один диск-учасник, швидкість читання більша принаймні в два рази, а якщо виробник RAID-контроллера порядний, то в чотири. Дозволяє вихід з ладу будь-якої кількості дисків, що залишає працездатними підмасиви RAID-1.

RAID-5 - оптимальне рішення при наявності трьох і більше дисків. Принцип роботи при наявності N дисків наступний: пишеться N-1 блок і контрольна сума. Контрольна сума пишеться кожен раз на інший фізичний диск. Таким чином швидкість запису зростає майже в N-1 разів (майже, бо крім запису потрібен час на обчислення контрольної суми). Швидкість читання зростає в N-1 раз. Розмір теж дорівнює сумарному розміру N-1 диска. Дозволяє вихід з ладу одного диска - загублені блоки відновлюються по контрольних сумах (правда в цьому випадку швидкість нижча ніж при читанні з одного диска, але інформація не втрачається).

RAID-6 - вимагає наявності чотирьох або більше дисків, від RAID-5 відрізняється наявністю двох контрольних сум, відповідно дозволяє вихід з ладу будь-яких двох дисків, але і показники швидкості читання/запису в N-2 рази більші і доступний розмір N-2 дисків-учасників. Що поробиш - за більшу надійність чимось треба жертвувати. При виході з ладу одного диска швидкість падає як в RAID-5, при виході з ладу двох дисків - просто дико падає, але дані на місці, а це найважливіше.

RAID-2,3,4 - теж існують, але тепер вже лише в теорії, оскільки в них, як і в RAID-5 обчислювалася контрольна сума, але під її збереження виділявся окремий диск, що призводило до його надактивного використання і швидкого виходу з ладу, що миттєво перетворювало їх на ненадійний RAID-0.

RAID-50 та **RAID-60** - теж теоретично можливі, але практична користь досить сумнівна, а алгоритм побудови дуже заплутаний, тому в реальному житті навряд чи хтось їх будує. При особливій паранойї можна задатися ще створенням RAID-51 та RAID-61, але це вже вершина збочення в RAID'обудуванні.

Найчастіше бюджетні материнські плати запропонують нам варіанти RAID-0, RAID-1, RAID-10, деякі - JBOD. Більш дорогі можуть також мати опції і RAID-5 та RAID-6. Щодо "інтегрованого RAID'обудування" слід пам’ятати одну корисну річ - у дуже багатьох випадках після задання конфігурації не відбувається абсолютно нічого, а фактично масив будується вже драйвером в опера-

ційній системі. Це так звані "фальшиві" RAID-контроллери, які всю свою роботу перекидають на процесор і пам’ять Вашого комп’ютера і, відповідно, відбирають їхні ресурси у Ваших задач. На відміну від них "чесні" RAID-контроллери мають власні процесор та пам’ять, але відповідно зростає і ціна такого технічного рішення. Так що Вам слід зважити: якщо Ваші задачі не надто навантажують процесор і пам’ять, але гальмують через повільну дискову систему, то можна обійтися і "фальшивкою", але якщо є потреба збільшити швидкість чи надійність дискових операцій, але процесор і пам’ять завантажені "під зав’язку", то доведеться розщедритися на неурізаний контроллер.

Якщо у Вас немає ні "фальшивого" ні "чесного" RAID-контроллера, то більшість операційок дозволяють створити RAID-масиви засобами самої системи - швидкість роботи, звичайно ж, така як і в "фальшивого" контролера, зате в опціях ми практично не обмежені.

Ще одна річ про яку слід знати пристубаючи до RAID'обудування, це можливість виділення spare-дисків - резервних дисків, які задіюються автоматично при виході з ладу допустимої кількості основних дисків і на які відбувається відновлення інформації та, відповідно, відновлення функціональності і швидкості RAID-масиву в початковому об’ємі.

Для RAID-5 та RAID-6 spare-диски є просто необхідними, якщо у Вас серйозна задача, адже при виході з ладу одного з дисків швидкість роботи цих масивів надзвичайно низька, а наявність такого диска дозволяє автоматично відновити на ньому дані з пошкодженого і повернутися до початкових швидкісних показників.

Spare-диски можна задати для всіх RAID'ів крім JBOD та RAID-0.

З теорією розібралися, тепер поглянемо де це все можна використати на практиці.

RAID-0 - улюблена забавка геймерів. Сучасні ігри використовують надзвичайно багато графічних текстур високої якості і чим шидше ці текстури будуть зчитуватися в пам’ять тим швидше забавка буде завантажуватися і менше гальмувати. Надійність є фактором некритичним - при "вильоті" диска масив можна переробити на меншу кількість, а ігри - перевстановити.

RAID-1 - використовується всюди, де присутня певна паранойя: коли серйозно розробляєте якісь проекти або ведете бухгалтерію, то збереження даних є на першому місці.

RAID-10 - потрібний тим, хто працює з мультимедійними проектами (відео, флеш-програмування), коли потрібно і дані не втратити і швидкість роботи збільшити.

RAID-5 та **RAID-6** - тоді ж коли і **RAID-10**, але хочеться максимально ефективно використати місце.

Модуль №3

Windows Live Safety Center. Microsoft Client Protection.

Технології, які пропонує компанія Microsoft для захисту від програм-шпигунів і вірусів: Windows Live Safety Center - веб-сервер-служба, що забезпечує нормальну роботу комп'ютера завдяки засобам сканування і видалення небажаних програм. Також дозволяє виконувати резервне копіювання файлів і дефрагментацію жорстких дисків; Microsoft Client Protection - засіб захисту комп'ютерів і файлових серверів від таких погроз, як програми-шпигуни і rootkit, а також від вірусів і інших традиційних способів атаки.

За останні кілька років корпорація Майкрософт доклала серйозних зусиль для вивчення шкідливих програм і розробки технологій, що дозволяють знизити рівень їх небезпеки. В рамках цих зусиль корпорація Майкрософт створила спеціальну службу по боротьбі з шкідливими програмами, в сферу дії якої входить вивчення шкідливих програм, програм-шпигунів і інших потенційно небажаних програм, а також випуск та обслуговування засоби видалення шкідливих програм для системи Windows (MSRT) і Windows Defender. Ця служба також розробляє основні технології боротьби з шкідливими програмами (включаючи модулі сканування і оновлення визначень шкідливих програм) для таких продуктів, як служба Microsoft ® Windows Live ™ OneCare, центр Windows Live Safety Center Beta, служби Microsoft Antigen, а також випуску Microsoft Client Security.

Корпорація Майкрософт випустила першу версію засоби MSRT 13 січня 2005 на 24 мовах для комп'ютерів з системами Windows ® 2000, Windows XP і Windows Server ™ 2003. Засіб призначений для виявлення і видалення найбільш поширених шкідливих програм з комп'ютерів клієнтів і надається користувачам ліцензійних систем Windows безкоштовно. На момент написання цього документа корпорація Майкрософт випустила 15 додаткових вдосконалених версій засоби і продовжує випускати нові версії у другій вівторок кожного місяця, додаючи в них додаткові визначення поширених шкідливих програм для виявлення і видалення. З моменту випуску першої версії засоби MSRT воно запускалося близько 2,7 мільярда раз більш ніж на 270 мільйонах комп'ютерів.

У даному звіті представлені докладні відомості про шкідливі програми, засновані на даних, зібраних засобом MSRT1, і відзначено вплив засоби MSRT на зниження активності шкідливих програм на комп'ютерах під управлінням системи Windows. Ключові висновки, зроблені на основі цих відомостей, наведені нижче. Детальніше вони представлені далі.

• За останні 15 місяців засіб MSRT видалило 16 мільйонів копій шкідливих програм з 5,7 мільйона комп'ютерів під управлінням системи Windows. В середньому це засіб видаляє одну копію шкідливої програми на кожних 311 комп'ютерах, на яких воно запускається.

• З 61 сімейства шкідливих програм, віддалених засобом MSRT з січня 2005 р. по лютий 2006 р., для 41 частота виявлення зменшилася після додаван-

ня їх в базу даних кошти. При цьому у 21 засоби спостерігалося зниження активності більш ніж на 75%.

• Програми-трояни, які дозволяють зловмиснику захоплювати контроль над зараженим комп'ютером і викрадати конфіденційну інформацію, - серйозна і відчутна загроза для користувачів системи Windows. Засіб MSRT видалило одну або декілька програм-троянів приблизно на 3,5 мільйона комп'ютерів.

Це означає, що програми-трояни присутні на 62% з 5,7 мільйона комп'ютерів, на яких засіб видаляло шкідливі програми. Більшість видалених програм були програмами-роботами, підвидом програм-троянів, які обмінюються даними за допомогою технології IRC.

• Програми типу Rootkit, які змінюють параметри системи, щоб приховати присутність інших,

можливо, шкідливих компонентів, є зростаючою потенційною загрозою, але на даний момент не дуже поширені. З 5,7 мільйона комп'ютерів, на яких засіб удаляло шкідливі програми, вони зустрічалися в 14% випадків. При цьому якщо виключити програму WinNT/F4IRootkit, поширювану з деякими музичними компакт-дисками фірми Sony, цей показник знижується до 9%. У 20% випадків при виявленні на комп'ютері програми типу rootkit там ж виявлялася і програма-троян.

• Серед випадків зараження шкідливими програмами серйозне місце займають атаки, засновані на методах «соціальної інженерії». Віруси-хробаки, поширювані по електронній пошті, через однорангові мережі і клієнти для обміну миттєвими повідомленнями, були видалені з 35% комп'ютерів, на яких працювало засіб.

• Проблема шкідливого програмного забезпечення має мігруючий характер. Більшість комп'ютерів, з яких кожна нова версія MSRT видаляє шкідливі програми, - це комп'ютери, на яких вони не були виявлені раніше. При роботі версії засоби MSRT за березень 2006 р. з'ясувалося, що з усіх заражених комп'ютерів лише близько 150 000 (20% від всіх комп'ютерів, на яких були виявлені шкідливі програми) вже були заражені раніше.

Огляд MSRT

Засіб видалення шкідливих програм (MSRT) призначено для виявлення і видалення найбільш поширених шкідливих програм з комп'ютерів клієнтів і надається користувачам ліцензійних систем Windows безкоштовно. Засіб MSRT поширюється головним чином через Центр оновлення Windows (WU), Служба Microsoft Update (MU) і за допомогою функції автоматичного оновлення (AU). Більш ранні версії засоби тепер доступні для завантаження на веб-вузлі Центру завантаження Microsoft, а також в якості елемента керування Microsoft ActiveX ® на веб-сайті

http://www.microsoft.com/malwareremove.

Поточна версія засобу здатна виявляти і видаляти 61 сімейство різних шкідливих програм.

Випускаючи даний засіб і здійснюючи його підтримку, корпорація Майкрософт переслідує дві основні мети:

1. Знизити рівень впливу поширених шкідливих програм на роботу користувачів системи Windows.

2. Використовувати дані, зібрані засобом MSRT, для отримання достовірних відомостей про шкідливих програмах, які на сьогоднішній день є серйозною перешкодою для користувачів системи Windows.

Ці дані використовувалися службою корпорації Майкрософт із боротьби з шкідливими програмами, щоб концентрувати зусилля по розробці і зменшити час, необхідний на усунення шкідливих програм.

Крім того, подібні звіти дозволяють іншим дослідникам використовувати містяться в них відомості для більш повного розуміння ситуації з шкідливими програмами, а також сконцентруватися на досягненні спільної мети - зниження впливу шкідливих програм на роботу користувачів системи Windows. Даний засіб не виявляє програми-шпигуни і інші небажані програми. Щоб виявляти і видаляти з комп'ютера програми-шпигуни і інші небажані програми, користувачам Windows необхідно завантажити та встановити новітні антишпигунські програми. Захисник Windows – анти шпигунське рішення корпорації Майкрософт, на даний момент знаходиться на стадії бета-версії. Користувачі справжньої ліцензійної версії системи Windows можуть завантажити його з веб-сторінки http://www.microsoft.com/windowsdefender.

Крім того, засіб MSRT не може замінити регулярно оновлюваний антивірусне рішення. Це пов'язано з відсутністю у даного засобу функції захисту в режимі реального часу, а також з тим, що воно використовує лише частина антивірусної бази даних сигнатур корпорації Майкрософт, яка дозволяє йому виявляти поширені шкідливі програми. Тим не менш, корпорація Майкрософт рекомендує користувачам, на комп'ютерах яких встановлені новітні антивірусні програми, також використовувати засіб MSRT в якості додаткової міри безпеки. Такі користувачі також отримають непряму користь від засобу MSRT, оскільки шкідливі програми можуть впливати на роботу спільно використовуваних ресурсів, наприклад Інтернету або локальної мережі.

Користувачам Windows настійно рекомендується встановити й регулярно оновлювати антивірусне рішення, надає захист в режимі реального часу і використовує повну версію бази даних сигнатур.

Служба Windows Live OneCare корпорації Майкрософт відповідає всім цим вимогам. Їм також відповідають і інші продукти, пропоновані партнерами корпорації Майкрософт, що займаються розробкою антивірусних програм. Повний список цих компаній можна знайти на веб-вузлі http:// www.microsoft.com / security / partners / antivirus.asp.

Модуль №4

Методи перевірки стану захищеності ОС

Методи перевірки стану захищеності ОС: Перевірка заголовків Banner check); Активні зондуючі перевірки (Active probing check); Імітація атак (Exploit check).

Перевірка заголовків "(banner check)

Зазначений механізм являє собою ряд перевірок типу "сканування" і дозволяє робити висновок про уразливість, спираючись на інформацію в заголовку відповіді на запит сканера. Типовий приклад такої перевірки - аналіз заголовків програми Sendmail або FTP-сервера, що дозволяє дізнатися їхню версію і на основі цієї інформації зробити висновок про наявність у них уразливості.

Найбільш швидкий і простий для реалізації метод перевірки присутності на сканованого вузлі уразливості. Однак за цією простотою ховається чимало проблем.

Ефективність перевірок заголовків досить ефемерною. І ось чому. По-перше, ви можете змінити текст заголовка, завбачливо видаливши з нього номер версії або іншу інформацію, на підставі якої сканер будує свої висновки. І хоча такі випадки виключно рідкісні, нехтувати ними не варто. Особливо в тому випадку, якщо у вас працюють фахівці в галузі безпеки, які розуміють всю небезпеку заголовків "за замовчуванням". По-друге, найчастіше, версія, що вказується в заголовку відповіді на запит, не завжди говорить про уразливість програмного забезпечення. Особливо це стосується програмного забезпечення, розповсюджуваного разом з вихідними текстами (наприклад, в рамках проекту GNU). Ви можете самостійно усунути уразливість шляхом модифікації вихідного тексту, при цьому забувши змінити номер версії в заголовку. І по-третє, усунення вразливості в одній версії ще не означає, що в наступних версіях ця уразливість відсутня.

Процес, описаний вище, є першим і дуже важливим кроком при скануванні мережі. Він не приводить до порушення функціонування сервісів або вузлів мережі. Однак не варто забувати, що адміністратор може змінити текст заголовків, що повертаються на зовнішні запити.

"Активні зондувальні перевірки" (active probing check)

Також відносяться до механізму "сканування". Однак вони засновані не на перевірках версій програмного забезпечення в заголовках, а на порівнянні "цифрового зліпка" (fingerprint) фрагмента програмного забезпечення зі зліпком відомої вразливості. Аналогічним чином поступають антивірусні системи, порівнюючи фрагменти сканованого програмного забезпечення з сигнатурами вірусів, що зберігаються в спеціалізованій базі даних. Різновидом цього методу є перевi-

рки контрольних сум або дати сканованого програмного забезпечення, які реалізуються в сканерах, що працюють на рівні операційної системи.

Спеціалізована база даних (в термінах компанії Cisco - база даних з мережевої безпеки) містить інформацію про вразливості і способах їх використанні (атаках). Ці дані доповнюються відомостями про заходи їх усунення, що дозволяють знизити ризик безпеки в разі їх виявлення. Найчастіше ця база даних використовується і системою аналізу захищеності і системою виявлення атак. Принаймні, так чинять компанії Cisco і ISS.

Цей метод також досить швидкий, але реалізується важче, ніж "перевірка заголовків".

"*Імітація атак*" *(exploit check)*

Перекладу терміна "exploit" в російських публікаціях я ніде не зустрічав і еквівалента в російській мові також не знайшов. Тому скористаюся перекладом "імітація атак". Дані перевірки відносяться до механізму "зондування" і засновані на експлуатації різних дефектів у програмному забезпеченні.

Деякі уразливості не виявляють себе, поки ви не "підштовхнете" їх. Для цього проти підозрілого сервісу або вузла запускаються реальні атаки. Перевірки заголовків здійснюють первинний огляд мережі, а метод "exploit check", відкидаючи інформацію в заголовках, дозволяє імітувати реальні атаки, тим самим з більшою ефективністю (але меншою швидкістю) виявляючи уразливості на сканованих вузлах. Імітація атак є більш надійним способом аналізу захищеності, ніж перевірки заголовків, і звичайно більш надійні, ніж активні зондувальні перевірки.

Однак існують випадки, коли імітація атак не завжди може бути реалізована. Такі випадки можна розділити на дві категорії: ситуації, в яких тест призводить до "відмови в обслуговуванні" аналізованого вузла або мережі, і ситуації, при яких уразливість в принципі не придатна для реалізації атаки на мережу.

Як ми всі знаємо, багато проблем захисту не можуть бути виявлені без блокування чи порушення функціонування сервісу або комп'ютера в процесі сканування. У деяких випадках небажано використовувати імітацію атак (наприклад, для аналізу захищеності важливих серверів), тому це може привести до великих витрат (матеріальним і тимчасовим) на відновлення працездатності виведених з ладу елементів корпоративної мережі. У цих випадках бажано застосувати інші перевірки, наприклад, активне зондування або, в крайньому випадку, перевірки заголовків.

Однак, є деякі уразливості (наприклад, перевірка схильності атакам типу "Packet Storm"), які просто не можуть бути протестовані без можливого виведення з ладу сервісу або комп'ютера. У цьому випадку розробники поступають таким чином, - за замовчуванням такі перевірки вимкнені і користувач може сам включити їх, якщо бажає. Таким чином, наприклад, реалізовані системи

CyberCop Scanner і Internet Scanner. В останній системі такого роду перевірки виділені в окрему категорію "Denial of service" ("Відмова в обслуговуванні"). При включенні будь-якої з перевірок цієї групи система Internet Scanner видає повідомлення "WARNING: These checks may crash or reboot scanned hosts" ("Увага: ці перевірки можуть вивести з ладу йди перезавантажити скановані вузли").

Етапи сканування

Практично будь-який сканер проводить аналіз захищеності у кілька етапів:

Збір інформації про мережу. На даному етапі ідентифікуються всі активні пристрої в мережі і визначаються запущені на них сервіси і демони. У разі використання систем аналізу захищеності на рівні операційної системи даний етап пропускається, оскільки на кожному аналізованому вузлі встановлені відповідні агенти системного сканера.

Виявлення потенційних вразливостей. Сканер використовує описану вище базу даних для порівняння зібраних даних з відомими уразливими за допомогою перевірки заголовків або активних зондувальних перевірок. У деяких системах всі уразливості ранжуються за ступенем ризику. Наприклад, у системі NetSonar уразливості діляться на два класи: мережеві та локальні уразливості. Мережеві уразливості (наприклад, впливають на маршрутизатори) вважаються серйознішими в порівнянні з уразливими, характерними тільки для робочих станцій. Аналогічним чином "надходить" і Internet Scanner. Всі уразливості в ньому діляться на три ступені ризику: високий (High), середня (Medium) і низька (Low).

Підтвердження обраних вразливостей. Сканер використовує спеціальні методи і моделює (імітує) певні атаки для підтвердження факту наявності вразливостей на обраних вузлах мережі.

Генерація звітів. На основі зібраної інформації система аналізу захищеності створює звіти, що описують виявлені вразливості. У деяких системах (наприклад, Internet Scanner і NetSonar) звіти створюються для різних категорій користувачів, починаючи від адміністраторів мережі і закінчуючи керівництвом компанії. Якщо перших в першу чергу цікавлять технічні деталі, то для керівництва компанії необхідно представити красиво оформлені із застосуванням графіків і діаграм звіти з мінімумом подробиць. Важливим аспектом є наявність рекомендацій щодо усунення виявлених проблем. І тут по праву лідером є система Internet Scanner, яка для кожної уразливості містить покрокові інструкції для усунення вразливостей, специфічні для кожної операційної системи. У багатьох випадках звіти також містять посилання на FTP-або Web-сервера, що містять patch'і і hotfix'и, що усувають виявлені вразливості.

Автоматичне усунення вразливостей. Цей етап дуже рідко реалізується в мережевих сканерах, але широко застосовується в системних сканерах (наприклад, System Scanner). При цьому дана можливість може реалізовуватися по-різному. Наприклад, в System Scanner створюється спеціальний сценарій (fix

script), який адміністратор може запустити для усунення вразливості. Одночасно зі створенням цього сценарію, створюється і другий сценарій, який скасовує зроблені зміни. Це необхідно в тому випадку, якщо після усунення проблеми, нормальне функціонування вузла було порушено. В інших системах можливості "відкату" не існує.

У будь-якому випадку у адміністратора, який здійснює пошук вразливостей, є кілька варіантів використання системи аналізу захищеності:

Запуск сканування тільки з перевірками на потенційні уразливості (етапи 1,2 і 4). Це дає попереднє ознайомлення з системами в мережі. Цей метод є набагато менш руйнівним в порівнянні з іншими і також є найшвидшим.

Запуск сканування з перевірками на потенційні і підтверджені уразливості. Цей метод може викликати порушення роботи вузлів мережі під час реалізації перевірок типу "exploit check".

Запуск сканування з вашими користувальницькими правилами для знаходження конкретної проблеми.

Все з вищезгаданого.

Особливості застосування

Якщо сканер не знаходить вразливостей на тестованому вузлі, то це ще не означає, що їх немає. Просто сканер не знайшов їх. І залежить це не тільки від самого сканера, але і від його оточення. Наприклад, якщо Ви тестуєте сервіс Telnet або FTP на віддаленій машині, і сканер повідомляє Вам, що вразливостей не виявлено - це може означати не тільки, що вразливостей немає, а ще й те, що на сканованого комп'ютері встановлений, наприклад, TCP Wrapper. Та хіба мало ще чого? Ви можете намагатися отримати доступ до комп'ютера через міжмережевий екран або спроби доступу блокуються відповідними фільтрами у провайдера і т.д. Для ОС Windows NT характерний інший випадок. Сканер намагається дистанційно проаналізувати системний реєстр (registry). Однак у разі заборони на аналізованому вузлі віддаленого доступу до реєстру, сканер ніяких вразливостей не виявить. Існують і більш складні випадки. І взагалі різні реалізації одного разом ж сервісу по-різному реагують на системи аналізу захищеності. Дуже часто на практиці можна побачити, що сканер показує уразливості, яких на аналізованому вузлі немає. Це відноситься до мережевих сканерів, які проводять дистанційний аналіз вузлів мережі. І віддалено визначити, чи існує в дійсності уразливість чи ні, практично неможливо. У цьому випадку можна порекомендувати використовувати систему аналізу захищеності на рівні операційної системи, агенти якої встановлюються на кожен контрольований вузол і проводять всі перевірки локально.

Для вирішення цієї проблеми деякі компанії-виробники пішли шляхом надання своїм користувачам декількох систем аналізу захищеності, що працюють на всіх зазначених вище рівнях, - мережевому, системному та рівні додатків.

Сукупність цих систем дозволяє з високим ступенем ефективності виявити практично всі відомі уразливості. Наприклад, компанія Internet Security Systems пропонує сімейство SAFEsuite, що складається з чотирьох сканерів: Internet Scanner, System Scanner, Security Manager і Database Scanner. На даний момент це єдина компанія, яка пропонує системи аналізу захищеності, що функціонують на всіх трьох рівнях інформаційної інфраструктури. Інші компанії пропонують або два (Axent) або, як правило, один (Network Associates, NetSonar та ін) сканер.

Компанія Cisco, яка пропонує тільки систему аналізу захищеності на рівні мережі пішла іншим шляхом для усунення проблеми помилкового спрацьовування. Вона ділить всі уразливості на два класи:

Потенційні - випливають з перевірок заголовків і т.зв. активних "підштовхування" (nudge) аналізованого сервісу або вузла. Потенційна уразливість можливо існує в системі, але активні зондувальні перевірки не підтверджують цього.

Підтверджені - виявлення та існуючі на аналізованому хості.

Перевірки на потенційну вразливість проводяться через колекцію заголовків і використання "несильних підштовхування". "Підштовхування" використовується для сервісів, які не повертають заголовки, але реагують на прості команди, наприклад, посилка команди HEAD для отримання версії HTTP-сервера. Як тільки ця інформація отримана, система NetSonar використовує спеціальний механізм (rules engine), який реалізує ряд правил, що визначають, чи існує потенційна уразливість.

Таким чином, адміністратор знає, які з виявлених вразливостей дійсно присутні в системі, а які вимагають підтвердження.

Проте в даному випадку залишаються уразливості, насилу виявляються або зовсім не виявляються через мережу. Наприклад, перевірка "слабкості" паролів, використовуваних користувачами та іншими обліковими записами. У разі використання мережевого сканера вам знадобиться затратити дуже багато часу на віддалену перевірку кожного облікового запису. У той же час, аналогічна перевірка, здійснювана на локальному вузлі, проводиться на кілька порядків швидше. Іншим прикладом може служити перевірка файлової системи сканованого вузла. У багатьох випадках її не можна здійснити дистанційно.

Переваги сканування на рівні ОС криються в прямому доступі до низькорівневих можливостям ОС хоста, конкретним сервісів і деталям конфігурації. Тоді як сканер мережевого рівня імітує ситуацію, яку міг би мати зовнішній зловмисник, сканер системного рівня може розглядати систему з боку користувача, вже має доступ до аналізованої системі і має в ній обліковий запис. Це є найбільш важливою відмінністю, оскільки мережевий сканер за визначенням не може надати ефективного аналізу можливих ризиків діяльності користувача.

Багато сканери використовують більш ніж один метод перевірки однієї і тієї ж уразливості або класу вразливостей. Однак у випадку великої кількості

перевірок використання кількох методів пошуку одній уразливості привносить свої проблеми. Пов'язано це зі швидкістю проведення сканування.

Наприклад, різниця між системами CyberCop Scanner і Internet Scanner в тому, що розробники з NAI ніколи не додадуть в свій продукт перевірку, якщо не можуть з упевненістю сказати, що перевірка надійно виявляє уразливість. В той час як розробники ISS поповнюють свою базу навіть у тому випадку, якщо їх перевірка виявляє уразливість з деякою точністю. Потім, уже після випуску системи, відбувається повернення до розробленим перевіркам, їх поліпшення, додавання нових механізмів здійснення перевірок тієї ж уразливості для підвищення достовірності, і т.д. Досить спірне питання, що краще. З одного боку краще, коли ви з упевненістю можете сказати, що на аналізованому вузлі певної уразливості немає. З іншого, навіть якщо існує хоч невеликий шанс, що ви можете виявити вразливість, то треба цим шансом скористатися. У будь-якому випадку найбільш кращим є перевірка типу "імітація атак", яка забезпечує найбільший відсоток точного виявлення вразливостей.

Не всі перевірки, розроблені в лабораторних умовах, функціонують так, як повинні. Навіть, незважаючи на те, що ці перевірки тестуються, перш ніж будуть внесені в остаточну версію сканера. На це можуть впливати деякі чинники:

Модуль №5

Система Kerberos. Служба автентифікації X.509. Захист електронної пошти S/MIME.

Керберос (протокол)

Протокол Кербер (Цербер, англ. Kerberos), орієнтований в основному на клієнт-серверну архітектуру, пропонує механізм взаємної аутентифікації двох співрозмовників (хостів) перед встановленням зв'язку між ними в умовах незахищеного каналу. Кербер - це також пакет вільного програмного забезпечення розробленого в Массачусетському технологічному інституті, що реалізовує цей протокол. Повідомлення протоколу Кербер захищені проти прослуховування мережі та повторних атак (англ. replay attack).

Кербер базується на симетричних алгоритмах шифрування та для своєї роботи потребує довірену третю сторону. Деякі модифікації протоколу можуть використовувати елементи асиметричного шифрування.

Історія і розвиток

Протокол Кербер розроблявся інститутом MIT розробив для забезпечення безпеки сервісів проекту Афіна, метою якого було забезпечення доступності навчальних матеріалів із будь-якої станції. Назва протоколу походить від грецької міфічної триголової потвори Цербера, захисника підземного царства. Існує декілька версій протоколу Включно із третьою були доступні лише для внутрішнього користування у MIT.

Стів Міллер (англ. Steve Miller) та Кліффорд Ньюман (англ. Clifford Neuman), основні архітектори четвертої версії Кербера, опублікували її у кінці 1980-х, хоча вона також розроблялась в основному для проекту Афіна. П'ята версія протоколу, розроблена Джоном Колем (англ. John Kohl) та Кліффордом Ньюманом, з'явилась у 1993 році як рекомендація на стандарт RFC 1510, оновлена 2005 року у RFC 4120.

Цілі

У своїй основі протокол Кербер ставить перед собою реалізацію таких принципів:

1. Пароль користувача ніколи не повинен передаватись по мережі;
2. Пароль користувача ні в якій формі не повинен зберігатись на клієнтській машині: він має бути ліквідований одразу після використання;
3. Пароль користувача не повинен зберігатись у незашифрованому вигляді навіть у базі даних аутентифікації (authentication server database);
4. Користувач вводить пароль лише раз за сесію. Таким чином, користувачі мають доступ до всіх сервісів, на які вони авторизовані, без

потреби заново вводити пароль під час сесії. Ця властивість також відома як Single Sign-On;

5. Управління аутентифікацією здійснюється централізовано сервером аутентифікації. Прикладні сервери, що надають послуги, не повинні містити аутентифікаційних відомостей користувачів. Це важливо для централізованого адміністрування облікових записів користувачів; не зберігається надлишкова інформація аутентифікації на різних серверах; при зміні користувачем паролю, він одночасно міняється для всіх наданих послуг.

Не лише користувачі зобов'язані підтвердити, що вони є тими, ким заявляють, але й прикладні сервери повинні підтвердити свою ідентичність користувачам. Цей процес називається Взаємна аутентифікація;

Після завершення етапів аутентифікації та авторизації, клієнт та сервер повинні мати можливість встановити зашифрований зв'язок. З цією метою Кербер підтримує генерацію і обмін ключів шифрування.

Опис протоколу

В основу Кербер покладений протокол Нідхема-Шредера. У ролі довіреної третьої сторони виступає Центр Розподілення Ключів (ЦРК, англ. Key Distribution Center), що складається із двох логічно розділених частин: Сервера Аутентифікації (ЦА, англ. Authentication Server) і Сервера Видачі Квитків (СВК, англ. Ticket Granting Server). Кербер працює на основі "квитків", які використовуються для підтвердження ідентичності користувачів.

ЦРК зберігає базу даних закритих ключів; закритий ключ учасника мережі відомий лише йому та ЦРК. Знання цього ключа є підтвердженням ідентичності учасника. Для зв’єднання двох учасників, ЦРК генерує ключ сесії, який забезпечує безпеку повідомлень. Безпека протоколу сильно залежить від синхронізації часу учасників мережі та від обмеження часу придатності квитків.

Спрощений опис протоколу виглядає таким чином

СА - Сервер Аутентифікації
СВК - Сервер Видачі Квитків
ПС - прикладний сервер, надає послуги
КВК - Квиток Видачі Квитків (англ. Ticket Granting Ticket)

Клієнт проходить аутентифікацію у СА з допомогою довготривалого спільного секрету і отримує квиток від СА. Пізніше клієнт використовує квиток для отримання додаткових квитків для ПС без потреби використання спільного секрету. Ці квитки підтверджують аутентифікацію для ПС.

X.509

У криптографії, X.509 — це стандарт ITU-T для інфраструктури відкритого (публічного) ключа (англ. public key infrastructure (PKI)) та інфраструктури управління привілеями (англ. Privilege Management Infrastructure (PMI)).

X.509 визначає стандартні формати для сертифікатів відкритого ключа (англ. public key certificates), списку відкликаних сертифікатів (англ. certificate revocation lists), атрибутів сертифікатів (англ. attribute certificates) та алгоритм валідації шляху сертифікації (англ. certification path validation algorithm).

Історія

X.509 був виданий 3 липня 1988 року і був зв'язаний зі стандартом X.500.

Особливості

Для випуску сертифікатів існує чітко визначена ієрархічна система відповідальних органів (англ. certificate authorities — CAs). У цьому його відмінність від моделей основаних на принципі мережі довіри (англ. web of trust), подібним до PGP, де будь-хто (не тільки спеціальні органи — CAs) можуть випускати, підписувати і перевіряти відповідність. Версія 3 X.509 має гнучкість для підтримки таких топологій як мости (bridges) та сітки (meshes). Може бути використаний у p2p мережах, але таким чином використовується рідко.

Структура сертифікату

Сертифікат (Certificate)
Версія (Version)
Серійний номер (Serial Number)
ідентифікатор алгоритму (Algorithm ID)
Видавець (Issuer)
Період дії (Validity) включає у себе:
не перед (Not Before)
не після (Not After)
предмет сертифікату (Subject)
інформація про предмет публічного ключа (Subject Public Key Info):
алгоритм (Public Key Algorithm)
предмет (Subject Public Key)
унікальний ідентифікатор видавця (Issuer Unique Identifier) — необов'язково
унікальний ідентифікатор предмету публічного ключа (Subject Unique Identifier) — необов'язково
розширення (Extensions)- необов'язково
...можливі додаткові деталі
алгоритм підпису сертифікату(Certificate Signature Algorithm)
підпис сертифікату(Certificate Signature)

Загальновживані розширення файлів сертифікатів

.CER — CER закодований сертифікат, або набір сертифікатів
.DER — DER закодований сертифікат

.PEM — (Privacy Enhanced Mail) Base64 закодований DER сертифікат, поміщений між «-----BEGIN CERTIFICATE-----» and «-----END CERTIFICATE-----»

.P7B — Див. .p7c

.P7C — PKCS#7 Підписана структура без даних, просто сертифікат(-и) чи список сертифікатів які вже не дійсні

.PFX — Див. .p12

.P12 — PKCS#12 можуть містити публічні та приватні ключі захищені паролем.

S/MIME

S / MIME (Secure / Multipurpose Internet Mail Extensions) - стандарт для шифрування і підпису в електронній пошті за допомогою відкритого ключа.

Призначення

S / MIME призначена для забезпечення криптографічного безпеки електронної пошти. Забезпечуються аутентифікація, цілісність повідомлення та гарантія збереження авторства, безпека даних (за допомогою шифрування). Велика частина сучасних поштових програм підтримує S / MIME.

Сертифікати S / MIME

Для використання S / MIME необхідно отримати і встановити індивідуальний ключ / сертифікат від центру сертифікації (ЦС). Краще всього використовувати різні ключі / сертифікати для цифрового підпису та шифрування, так як це дозволить розкрити за певних умов ключ шифрування (наприклад, за рішенням суду), не дискредитуючи при цьому цифрові підписи. Для шифрування повідомлення потрібно знати сертифікат прийомної сторони, що зазвичай забезпечується автоматично при одержанні листа з сертифікатом. Хоча технічно можливо послати повідомлення, зашифроване сертифікатом одержувача і не підписувати повідомлення власним, наприклад, через відсутність його, на практиці, програми з підтримкою S / MIME зажадають встановлення сертифіката відправника перед тим як дозволити шифрування повідомлень.

Звичайний основний особистий сертифікат засвідчує ідентичність власника тільки шляхом зв'язування воєдино поштової адреси та сертифіката. Він не засвідчує ні ім'я, ні рід діяльності. Більш повне посвідчення можна отримати, звернувшись до спеціалізованих ЦС, які надають додаткові (нотаріально еквівалентні) послуги або безпечну інфраструктуру відкритого ключа.

В залежності від політик ЦС, ваш сертифікат і весь його вміст можуть бути відкрито опубліковані для ознайомлення і перевірки. У такому випадку, ваше ім'я та поштову адресу стають доступними для всіх, в тому числі і для пошуку. Інші ЦС можуть публікувати лише серійні номери й ознака відкликання. Це необхідний мінімум для забезпечення цілісності інфраструктури відкритого ключа.

Перешкоди при практичному використанні S / MIME

Не всі програми електронної пошти можуть обробляти S / MIME, що призводить до листів із доданим файлом «smime.p7s», що може призвести до непорозуміння.

Іноді вважається, що S / MIME не сильно підходить для використання вебпошти. Так як вимоги безпеки вимагають, щоб сервер ніколи не зміг отримати доступ до закритого ключа, що зменшує таку перевагу вебпошти, як доступність з будь-якої точки.

Багато розрізняють закриті ключі для розшифровки і для цифрового підпису. Тих, хто готовий надати деякого агенту перший набагато більше, ніж тих, хто готовий надати другий. Якщо необхідно безпечне підтвердження авторства (як і забезпечення відсутності помилкового підтвердження), то другий ключ повинен бути під суворим контролем власника, і тільки його, протягом усього циклу його життя, від створення, до знищення.

S / MIME спеціально призначене для забезпечення безпеки на шляху від відправника до по-одержувача. Однак шкідливе ПЗ може потрапити в лист ще на стороні відправника при складанні листа, тоді воно без перешкод дійде до одержувача. Отже, необхідно забезпечувати безпеку на крайовому пристрої.

Література

1. В. Иллингуорт и др. Толковый словарь по вычислительным системам. М.: Машиностроение, 1991. - 560 с.
2. Елманова Н. Виртуальные машины и средства их создания. Часть 1. Microsoft Virtual PC 2004. КомпьютерПресс. №8. 2004. С. 158-160.
3. Jerry Honeycutt Microsoft Virtual PC 2004 Technical Overview.
4. Елманова Н., Пахомов С. Виртуальные машины 2007. КомпьютерПресс. №9. 2007. С.29-41.
5. Microsoft Windows XP Professional. Учебный курс MCSA/MCSE. М.: Русская редакция, 2003. - 1008 с.
6. Безмалый А. Централизованная замена системного ПО с помощью Microsoft BDD
7. Бейкер Дон, Автоматическое развертывание систем Windows 2000.
8. Васьков А., Пошаговое руководство по удаленной установке ОС.
9. Галатенко В.А. Основы информационной безопасности. М.: Изд-во ИНТУИТ.ру, 2005. - 208 с
10. Гапанович А. Платформа 2005: использование Business Desktop Deployment для автоматизации типовых задач администрирования.
11. Дуглас Стин, Использование решения для развертывания настольных бизнес-систем в малых и средних организациях.
12. Карпюк В.В. Microsoft Windows XP Professional. Опыт сдачи сертификационного экзамена 70-270. Спб.: БХВ-Петербург, 2004. - 528 с.
13. Галатенко В.А. Основы информационной безопасности. М.: Изд-во ИНТУИТ.ру, 2005. - 208 с.
14. Закер Крейг, Официальный учебный курс Microsoft: Управление и поддержка Microsoft Windows Server 2003 (70-290) М.: ЭКОМ; БИНОМ. Лаборатория знаний, 2006. - 447 с.
15. Использование программы "Архивация данных" в Microsoft Windows XP.
16. Холме Дэн, Томас Орин, Управление и поддержка Microsoft Windows Server 2003. Учебный курс MCSA/MCSE, М.: Русская редакция, 2004. - 448 с.
17. Елманова Н. Инструменты Microsoft для защиты от вредоносного ПО: ближайшее будущееКомпьютерПресс. №5. 2006. С. 162–164.
18. Домашняя страница Защитника Windows.
19. Windows Defender: System requirements.
20. Касперская Н. Безопасность от Microsoft: шаг к обновленному миру?
21. Frequently asked questions about Windows Defender.
22. Microsoft Baseline Security Analyzer.
23. Лукацкий А.В. Как работает сканер безопасности?
24. Семёнов В. Сканер уязвимостей XSpider 7
25. Средство Microsoft Baseline Security Analyzer (MBSA) версии 1.2.1
26. Феллинг Д. Microsoft Baseline Security Analyzer
27. Хоуи Д. Обеспечение безопасности с помощью MBSA 2.0