**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ
МІЖНАРОДНИЙ ЕКОНОМІКО-ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
ІМЕНІ АКАДЕМІКА СТЕПАНА ДЕМ'ЯНЧУКА**

І.Ф.ЧЕРНЕЦЬКИЙ

ТЕХНОЛОГІЇ КОМП'ЮТЕРНОЇ БЕЗПЕКИ

Книга 5

**Науковий керівник:
Р.М.Літнарович, доцент,к.т.н.**

Рівне, 2011

УДК 614.2 Чернецький І.Ф. Технології комп’ютерної безпеки. Монографія. Книга 5.Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2011.-95 с. Chernetskiy I.F. Technologies of computer safety. Book 5. Monograph. IEGU, Rivne, 2011.-97 p.
Рецензенти: В.Г.Бурачек, доктор технічних наук, професор
Є.С. Парняков, доктор технічних наук, професор
В.О.Боровий , доктор технічних наук, професор
Відповідальний за випуск: Й.В. Джунь, доктор фізико-математичних наук, професор
Послідовно розглядаються основні поняття побудови сучасних технологій комп’ютерної безпеки. Монографія містить актуальний матеріал довідково-аналітичного характеру по наступних темах: основи безпеки даних в комп’ютерних системах, ідентифікація і аутентифікація користувачів, захист даних від несанкціонованого доступу, основи захисту даних від комп’ютерних вірусів, основи криптографії, криптографічні методи захисту інформації, стандарти захисту інформації.
Ключові слова: комп'ютерна безпека, інформаційна безпека, захист, інформація.
Последовательно рассматриваются основные понятия построения современных технологий компьютерной безопасности. Монография содержит актуальный материал справочно аналитического характера по следующим темам: основы безопасности данных в компьютерных системах, идентификация и аутентификация пользователей, защита данных от несанкционированного доступа, основы защиты данных от компьютерных вирусов, основы криптографии, криптографические методы защиты информации, стандарты защиты информации.
Ключевые слова: компьютерная безопасность, информационная безопасность, защита, информация.
The basic concepts of construction of modern technologies of computer safety are consistently examined. A monograph contains aktual material certificate analytical character on the followings themes: bases of safety of information in the computer systems, authentication and authentification of users, protection of data from an unauthorized division, bases of protection of data from computer viruses, bases of cryptography, cryptographic methods of priv, standards of priv.
Keywords: computer safety, informative safety, defence, information

Чернецький Іван Федорович спеціаліст системотехнік, магістрант

інформаційних технологій

Зміст

Вступ

Люди, йдучи з дому, зазвичай закривають вхідні двері на замок. Вони також замикають свої автомобілі, залишаючи їх припаркованими на вулиці або на стоянці. І як правило, не повідомляють номер своєї кредитної картки першому зустрічному співрозмовнику, який пристає до перехожих, і настирливо пропонуючи купити в нього товари сумнівної якості. Однак переважна більшість людей до кінця не усвідомлює, наскільки сильно вони ризикують, якщо не піклуються про захист інформації, що знаходиться в їхніх комп'ютерах. Достовірно відомо, що лише окремі користувачі вживають хоч якісь заходи, покликані зберегти їхні дані. Решта всерйоз замислюються про це тільки тоді, коли втрачають інформацію, збережену в комп'ютері. Більш того, їхні комп'ютерні системи найчастіше зовсім не захищені від крадіжок і вандалізму. Кожен раз, використовуючи свій комп'ютер, його власник додає туди певну порцію інформації. Саме ця сукупна інформація і є найбільш цінним компонентом всієї комп'ютерної системи. А це означає, що якщо не вжити спеціальних заходів для її захисту, витрати, які понесе користувач, спробувавши відновити втрачені дані, значно перевищать вартість апаратних засобів, що використовуються для зберігання цих даних. Ще більш загрозливою небезпечними наслідками є ситуація, при якій податкова і банківська інформація користувача або його ділова переписка потрапляє в чужі руки. Важко собі уявити, що хтось, перебуваючи при здоровому розумі і твердій пам'яті, з доброї волі надає свою особисту інформацію людям, з якими не має або не бажає мати жодних справ. Але навіть якщо вам нема чого приховувати від сторонніх (у це важко повірити, але припустимо, що це дійсно так), неодмінно знайдеться хтось, хто не проти перетворити ваш комп'ютер в купу непотрібного мотлоху, як тільки випаде така можливість. Розплодилися киберпанки, крекери, фрікерів і брейкери з великим ентузіазмом займаються електронним злодійством, що можна порівняти з осквернення могил і розмальовування стін будинків непристойними написами. Не слід нехтувати і захистом даних від стихійних біду. Адже зовсім не важливо, зіпсується ваш жорсткий диск від комп'ютерного вірусу, від рук злісного хакера, або ту ж саму ведмежу послугу вам надасть ураган, повінь або кульова блискавка.
Не менш важливо відзначити, що незалежно від того, наскільки цінна ваша інформація, українським законодавством вона безумовно визнається об'єктом вашої власності. І ви, як власник своєї інформації. маєте право визначати правила її обробки та захисту. Базовим у цьому відношенні Державний стандарт України із захисту Інформації який може вживати необхідних заходів для запобігання витоку, розкрадання, втрати, спотворення і підробки інформації. Питання полягає в тому, які дії є насправді необхідними для адекватного захисту вашої інформації. Ризикну припустити, що ви навряд чи залишаєте свій будинок навіть на короткий час без того, щоб не замкнути

двері, хоча це досить клопітке заняття. По-перше, необхідно володіти мінімумом технічних знань, щоб підібрати і встановити надійний замок. По-друге, потрібен постійний контроль за станом замку, щоб утримувати його в справності. По-третє, щоб замок запобігав проникнення в будинок сторонніх людей, ви повинні дотримуватися певні правила (зберігати ключі в надійному місці, а також не залишати двері незамкненими). Подібні ж правила застосовні і у випадку захисту інформації в комп'ютерних системах. Саме тому так важливо відшукати розумний компроміс між цінністю ваших даних і незручностями, пов'язаними з використанням необхідних заходів безпеки. Як і дверний замок, будь-яка система комп'ютерного захисту інформації не є цілком безпечною. Завжди знайдеться хтось, здатний зламати захисні механізми комп'ютера. На щастя, такі люди зустрічаються дуже рідко, інакше єдиний спосіб захистити наші дані полягав би у їх знищення. Таким чином, завдання забезпечення інформаційної безпеки суперечлива за самою своєю суттю. З одного боку, засобів забезпечення безпеки ніколи не буває занадто багато в тому сенсі, що захист завжди можна тим або іншим способом подолати (просто кожен раз, коли підвищується рівень захисту, доводиться вигадувати більш витончений спосіб її обходу). З іншого боку, чим сильніше когось або щось захищають, тим більше виникає незручностей і обмежень, і в результаті замість почуття спокою інформаційна безпека викликає лише роздратування і прагнення від неї відмахнутися, як від набридливої мухи. Наприклад, за рахунок жорсткого контролю за доступом до комп'ютерної системи за допомогою паролів безсумнівно знижується ймовірність їх підбору зломщиком, проте одночасно це змушує рядових користувачів докладати значно більше зусиль для придумування і запам'ятовування паролів. А установка строгих обмежень на доступ до інформації створює додаткові труднощі при спільній роботі з цією інформацією. Тому ідеальною та універсальної системи захисту інформації не існує: тут все дуже індивідуально, і варіант захисту, найбільш близький до оптимального, весь час доводиться підбирати заново. Уважно вивчивши пропоновану вашій увазі книгу, ви зможете підібрати оптимальні методи для захисту комп'ютерної інформації. При цьому дуже істотну роль гратиме ступінь ізольованості вашого комп'ютера від зовнішнього світу. Якщо комп'ютер знаходиться у вас вдома або в офісі і фізично не пов'язаний з іншими комп'ютерами (за допомогою модему або апаратної мережі), то, цілком ймовірно, найбільша небезпека полягає в тому, що хтось занесе в нього вірус чи троянську програму. Друга, вкрай несприятлива перспектива, полягає в тому, що ваш комп'ютер може бути викрадений. Решту неприємностей можна уникнути, застосовуючи парольний захист комп'ютера і шифруючи файли, що містять конфіденційну інформацію. У цьому випадку забезпечити свій комп'ютер досить надійним захистом зможе будь-яка людина, навіть не дуже сильний в питаннях інформаційної безпеки. Якщо комп'ютерна система підключена до мережі, то потрібно прийняти додаткові заходи безпеки, кваліфіковано встановити і правильно

використовувати які під силу тільки підготовленим спеціалісту, який має досвід роботи в галузі захисту інформації. Зашиті механізми, вбудовані в мережні операційні системи, всілякі брандмауери та аналізатори безпеки мереж вимагають дуже гонкою па-будівництва. Тому навіть незначна помилка при їх установці і настройці чревата серйозними наслідками, оскільки зловмиснику досить виявити лише одне слабке місце в системі захисту, щоб здійснити її злом. Однак і у випадку, якщо комп'ютер працює і складі мережі, вивчення пропонованої книги буде корисно всім - від технаря до менеджера, не має глибоких знань в цих питаннях. Адже правильне використання засобів захисту інформації можливе лише в тому випадку, коли всі люди, так чи інакше причетні до роботи з середи вами обробки та зберігання даних, усвідомлюють (нехай навіть у найзагальніших рисах) принципи забезпечення надійного функціонування цих засобів і скрупульозно слідують даними принципам на практиці.

1. Загрози комп'ютерної безпеки

1.1.Проблеми правового захисту комп’ютерної інформації в Україні

Проблеми правового захисту комп’ютерної інформації зацікавили юристів провідних зарубіжних країн у 70-80-ті рр., коли почався розквіт комп’ютерної техніки. Поява і розповсюдження в 70-ті рр. компактних і порівняно недорогих персональних комп’ютерів створили можливість підключення до потужних інформаційних потоків необмеженого кола осіб, призвели до комп’ютеризації господарської і управлінської діяльності, використання комп’ютерної техніки в космічних дослідженнях, обороні, атомній енергетиці та інших сферах життя суспільства, де порушення роботи такої техніки здатне викликати аварії і навіть катастрофи з людськими жертвами і величезними економічними втратами. Крім того, поява комп’ютерних банків з інформацією персонального характеру робить неправомірний доступ до неї вельми небезпечним для прав і свобод людини. Постало питання про контрольованість доступу до інформації та її збереження. Особливо гостро така проблема стоїть у країнах з високорозвиненими технологіями й інформаційними мережами. Традиційні заходи (організаційні, програмні, технічні) не можуть повною мірою відігравати роль стримуючого фактора. У зв’язку з цим велика увага приділяється розвитку кримінального законодавства. Під кримінально-правовими заходами боротьби з комп’ютерною злочинністю в літературі розуміють прийняття кримінально-правових норм, якими встановлюється кримінальна відповідальність за вчинення окремих діянь у сфері використання комп’ютерної техніки. Зіткнувшись з комп’ютерною злочинністю, правоохоронні органи спочатку вели з нею боротьбу з допомогою традиційних правових норм про викрадення, привласнення, шахрайство і т.д. Однак, такий підхід виявився не зовсім вдалим, оскільки багато комп’ютерних злочинів не охоплюються складами традиційних

злочинів. Невідповідність кримінологічної реальності і кримінально-правових норм вимагали розвитку останніх. Цей розвиток відбувається у двох напрямках: 1) ширше трактування традиційних норм і їх застосування за аналогією;2) розробка спеціалізованих норм про комп'ютерні злочини. Що ж таке комп'ютерний злочин? Деякі вчені піддають сумніву доцільність вживання терміна "комп'ютерні злочини". Наприклад, В.В.Крилов пропонує як альтернативне ширше розуміння – "інформаційні злочини", яке дозволяє абстрагуватися від конкретних технічних засобів. М.Ф.Ахраменко під комп'ютерним злочином розуміє «вчинення винного суспільно-небезпечного протиправного діяння з використанням інформаційно-обчислювальних систем або із впливом на них». Кримінальна поліція Німеччини взяла на озброєння визначення комп'ютерної злочинності, яке включає в себе «всі протизаконні дії, за яких електронна обробка інформації була знаряддям їх скоєння або об'єктом». Швейцарські експерти під комп'ютерною злочинністю розуміють «всі зумисні та протизаконні дії, які призводять до нанесення шкоди майну і скоєння яких стало можливим завдяки електронній обробці інформації». У 1973 р. у Швеції приймається закон, згідно з яким встановлена відповідальність за неправомірну зміну, знищення або доступ до записів на комп'ютерних носіях (інформаційні зловживання). Пізніше спеціальні норми про комп'ютерні злочини були прийняті в США, Великобританії, Австрії, Канаді (липень 1985 р.), Данії (грудень 1985 р.), Австралії, Франції, Португалії (1982 р.) й інших країнах. Необхідність видання законів, спеціально орієнтованих на боротьбу з комп'ютерними злочинами, досить швидко була усвідомлена в США на рівні законодавчих органів окремих штатів. На початок 70-х рр. відповідні закони були видані в шести штатах, а робота над законопроектами велася у дванадцяти інших. До 1985 р. такі акти були прийняті в 47 штатах. Наприклад, у штаті Флорида "Закон про комп'ютерні злочини" набув сили 1 січня 1978 р. і є найбільш грунтовним з усіх аналогічних актів інших штатів США. Згідно з даним законом конкретні види комп'ютерних злочинів розподілені на три групи:

• злочини проти інтелектуальної власності, тобто зумисне незаконне внесення змін, знищення або викрадення даних, програм і документації, пов'язаної з комп'ютерами;

• злочини, що завдають шкоди комп'ютерному обладнанню, тобто знищення або пошкодження комп'ютерних систем, приладів і т.д.;

• злочини проти користувачів комп'ютерів, тобто, будь-яке незаконне використання чужого комп'ютера, в тому числі спроба обробити на ньому які-небудь дані, недопущення до користування комп'ютером особи, яка має на це право. На думку автора, до числа комп'ютерних злочинів доцільно віднести як злочини у сфері комп'ютерної інформації, так і злочини, які вчиняються з використанням комп'ютерних технологій. Комп'ютерною визнається інформація, тобто, відомості про осіб, предмети, факти, події, явища і процеси, що зберігаються в комп'ютерній системі, мережі або на

машинних носіях. Правове забезпечення комп'ютерної безпеки – сукупність норм права, що визначають суспільні відносини, які виникають у процесі діяльності людей щодо безпечного використання комп'ютерної техніки для обробки інформації. Ці правові норми можуть бути представлені у вигляді законів, положень, інструкцій, інших нормативно-правових документів.
За В.І.Ярочкіним , предметом правового регулювання в області гарантування інформаційної, в тому числі комп'ютерної безпеки є:

• правовий режим інформації, засобів інформатики, індустрії інформатизації і систем інформаційних послуг в умовах ризику, засоби і форми захисту інформації.

• правовий статус учасників правовідносин у процесах інформатизації;

• порядок стосунків суб'єктів з урахуванням їх правового статусу на різних стадіях і рівнях процесу функціонування інформаційних структур і систем. О.П.Крюкова під правовим забезпеченням інформаційної безпеки розуміє:

• захист інтересів фізичних осіб – шляхом введення норм, які встановлюють межі збирання і використання відомостей про цих осіб з боку держави або інших суб'єктів;

• захист інтересів держави і суспільства – шляхом встановлення пріоритетів захисту інформації, яка охороняється як власність держави;

• захист інтересів юридичних і фізичних осіб – шляхом встановлення норм, які регулюють поводження з інформацією, що охороняється і забезпечує діяльність цих осіб, і встановлення механізмів захисту цих суб'єктів. З метою уніфікації національних законодавств у 1989 році комітетом міністрів Європейського союзу був узгоджений і затверджений Список правопорушень, рекомендований країнам-членам ЄС з метою розробки єдиної кримінальної стратегії, пов'язаної з комп'ютерними злочинами. "Мінімальний список правопорушень" містить наступні вісім видів комп'ютерних злочинів:

1. Комп'ютерне шахрайство. Введення, зміна, стирання або пошкодження даних ЕОМ чи програм ЕОМ, або ж інше втручання в хід обробки даних, яке впливає на хід обробки даних таким чином, що служить причиною економічних втрат або викликає втрату майна іншої людини з наміром незаконного покращення економічного становища для себе або іншої особи (як альтернатива – з наміром до незаконного позбавлення цієї особи її майна).

2. Підробка комп'ютерної інформації. Несанкціоноване стирання, пошкодження, погіршення або пригнічення даних ЕОМ або інше втручання в хід обробки даних різними способами, або створення таких умов, які згідно з національним законодавством складатимуть таке правопорушення, як підробка в традиційному розумінні такого порушення.

3. Пошкодження даних ЕОМ або програм ЕОМ. Несанкціоноване стирання, пошкодження або пригнічення даних ЕОМ або програм ЕОМ.

4. Комп'ютерний саботаж. Введення, зміна, стирання, пошкодження даних ЕОМ або втручання в системи ЕОМ з наміром перешкоджання функціонуванню комп'ютера або системи передачі даних.

5. Несанкціонований доступ. Несанкціонований доступ до системи ЕОМ через мережу з порушенням засобів захисту.

6. Несанкціоноване перехоплення даних. Несанкціоноване перехоплення з допомогою технічних засобів зв'язку як у межах комп'ютера, системи або мережі, так і ззовні.

7. Несанкціоноване використання захищених комп'ютерних програм. Незаконне відтворення, розповсюдження або зв'язок з програмою ЕОМ, яка захищена у відповідності з законом.

8. Несанкціоноване відтворення схем. Несанкціоноване відтворення схемних рішень, захищених у відповідності з законом про напівпровідникові вироби (програми), або комерційна експлуатація, або незаконне імпортування з цією ж метою схеми або напівпровідникового виробу як продукту, створеного з використанням даних схем. "Необов'язковий список порушень" включає в себе наступні чотири види комп'ютерних злочинів:

1. Зміна даних ЕОМ або програм ЕОМ. Незаконна зміна даних або програм ЕОМ.

2. Комп'ютерний шпіонаж. Придбання з використанням незаконних засобів або шляхом несанкціонованого розкриття, пересилання або використання торгових або комерційних таємниць з допомогою подібних методів чи інших незаконних засобів з тим чи іншим наміром, що завдає економічної шкоди особі шляхом доступу до його таємниць або дозволяє отримати незаконну економічну перевагу для себе чи іншої особи.

3. Використання ЕОМ без дозволу. Використання системи ЕОМ або комп'ютерної мережі без відповідного дозволу є злочинним, коли воно:

• інкримінується в умовах великого ризику втрат, викликаних невідомою особою, яка використовує систему або завдає шкоди системі чи її функціонуванню;
• інкримінується невідомій особі, яка має намір завдати шкоди і використовує для цього систему або завдає шкоди системі чи її функціонуванню;
• застосовується у випадку, коли втрачається інформація з допомогою невідомого автора, який використав дану систему або завдав шкоди системі чи її функціонуванню.

4. Використання захищеної програми ЕОМ без дозволу.

Використання без дозволу захищеної програми ЕОМ або її незаконне відтворення з наміром виправити програму таким чином, аби отримати незаконну економічну вигоду для себе або іншої особи або завдати шкоди законному власникові даної програми. Протягом останніх десятиліть в США прийнято ряд федеральних законів, що створили правову основу для формування і проведення єдиної державної політики в області інформатизації і захисту інформації. Наприклад, закон Сполучених Штатів "Про забезпечення безпеки ЕОМ" № HR 145, прийнятий конгресом у травні 1987 року, встановлює пріоритет національних інтересів при вирішенні питань безпеки інформації, у тому числі й приватної інформації. Законом встановлено, що "важливою" є така інформація, "втрата якої, неправильне використання, несанкціонована зміна якої або доступ до якої можуть призвести до небажаного впливу на національні інтереси". Встановлена також категорія інформації обмеженого доступу – "нетаємна, але важлива з точки зору національної безпеки". До цієї категорії віднесена переважна частина відомостей, що циркулюють або обробляються в інформаційно-телекомунікаційних системах приватних фірм і корпорацій, які працюють за державним замовленням. Комп'ютерні злочини у США можуть підпадати як під юрисдикцію штату, в якому вони скоєні, так і під федеральні закони. Порушення федерального законодавства відбувається у випадках, коли:

• злочин скоювався стосовно комп'ютерних систем, що належать урядові США;

• злочин скоювався стосовно комп'ютерних систем, що належать банкам або іншим фінансовим організаціям;

• відбулося викрадення або розкриття інформації про національну оборону, атомну енергетику, іншої державної закритої інформації;

• злочин скоювався з інших штатів або держав;

• при скоєнні злочину використовувалися міжнародні системи зв'язку.

Сьогодні в світі близько 20 країн мають національне законодавство, що стосується використання глобального інформаційного простору . До розряду пріоритетних висувається питання правових і організаційних механізмів регулювання використання Internet. В Internet відсутня централізована система управління. Координатором виступає Товариство учасників Internet (ISOC), яке є громадською організацією, що базується на внесках і пожертвуваннях спонсорів. Зарубіжними дослідниками неодноразово підкреслювалась необхідність не обтяжувати Internet зайвим державним регулюванням. Однак, розвиток цієї глобальної мережі ставить ряд правових проблем, для вирішення яких уже прийняті і готуються до прийняття низка законодавчих актів. Одним з важливих кроків, спрямованих на врегулювання цієї проблеми, є прийняття Радою Європи (Council of Europe) 23 листопада 2001 року Конвенції про кіберзлочинність. Враховуючи складність проблеми, Рада Європи підготувала й опублікувала проект Конвенції щодо боротьби зі злочинами в кіберпросторі ще на початку 2000 року. Цей документ став

першою міжнародною угодою з юридичних і процедурних аспектів розслідування і кримінального переслідування кіберзлочинів. Конвенцією передбачаються скоординовані на національному і міждержавному рівнях дії, спрямовані на недопущення несанкціонованого втручання в роботу комп'ютерних систем, незаконного перехоплення даних і втручання в комп'ютерні системи. Прийняття державами-членами Ради Європи "Конвенції про кіберзлочинність" стало результатом розуміння важливості проведення політики, направленої на захист суспільства від кіберзлочинів, необхідності появи відповідного законодавства і зміцнення міжнародного співробітництва. Одним із головних висновків, який можна зробити, аналізуючи "Конвенцію", є вироблення спільної позиції з питання про те, які діяння, пов'язані з використанням комп'ютерних систем, мають бути криміналізовані. З правової точки зору велике значення мають загальні принципи, що стосуються міжнародного співробітництва. Це питання видачі комп'ютерних злочинців і надання один одному взаємодопомоги при розслідуванні кримінальних справ, пов'язаних з комп'ютерними системами і даними. З урахуванням специфіки соціального феномену кіберзлочинності, масштабів інформатизації і розвитку глобальної мережі Інтернет стає все менш вірогідним, що подібні злочини обмежуватимуться територією однієї держави. У процесі проведення розслідування правоохоронні органи різних держав мають співробітничати між собою, надаючи потенційно корисну інформацію один одному. В зв'язку з правовою допомогою, при розслідуванні кіберзлочинів неодмінно виникатимуть і інші проблеми. Якщо внутрішнім правом однієї з сторін не передбачені конкретні повноваження щодо пошуку доказів в електронному середовищі, така сторона буде не в змозі адекватно реагувати на прохання про надання допомоги. З цієї причини важливою умовою міжнародного співробітництва є узгодження повноважень вживати необхідних заходів для розслідування таких видів злочинів. Обмеженість національного законодавства і відсутність єдиної правової бази правоохоронних органів у боротьбі з даним видом правопорушень – ось одна з головних причин зростання кількості комп'ютерних злочинів. Лише шляхом органічного поєднання кримінально-правових і криміналістичних стратегій боротьби з даним видом злочинів можна досягти успіху. Важливою складовою такої стратегії має стати міжнародне співробітництво в даній сфері, оскільки вже очевидно, що контролювати транснаціональну складову кіберзлочинів на рівні окремих держав практично неможливо. Міжнародне співробітництво в боротьбі зі злочинністю в сфері використання комп'ютерних технологій потребує правового, організаційного і наукового забезпечення. Підписання "Конвенції" в Будапешті 23 листопада 2001 року главою української делегації Сюзанною Станик, безперечно, сприятиме зміцненню міжнародного співробітництва в боротьбі з кіберзлочинністю. Нещодавно Президентом України підписані прийняті парламентом зміни до Концепції національної безпеки України. В Концепції уточнюється перелік пріоритетів національних інтересів України, до яких, зокрема, належать:

гарантування конституційних прав і свобод людини і громадянина, захист державного суверенітету, територіальної цілісності і недоторканості кордонів, невтручання іноземних держав у внутрішні справи країни, створення конкурентоспроможної, соціально орієнтованої ринкової економіки, забезпечення постійного зростання рівня життя і добробуту населення. Визначаються загрози національній безпеці та інтересам України, серед яких комп'ютерна злочинність і комп'ютерний тероризм належать до числа пріоритетних. До основних принципів, за якими формується і проводиться державна політика України у сфері захисту інформації належать, перш за все: - додержання балансу інтересів особи, суспільства та держави, їх взаємна відповідальність; - єдність підходів до забезпечення захисту інформації, які визначаються загрозами безпеці інформації та режимом доступу до неї; - комплексність, повнота та безперервність заходів захисту інформації; - відкритість нормативно-правових актів та нормативних документів з питань захисту інформації, які не містять відомостей, що становлять державну таємницю; - узгодженість нормативно-правових актів організаційно-управлінського змісту та нормативних документів з питань технічного захисту інформації з відповідними міжнародними договорами України; - обов'язковість захисту інженерно-технічними заходами інформації, що складає державну та іншу, передбачену законом, таємницю, конфіденційної інформації, що є власністю держави, відкритої інформації, важливої для держави, незалежно від того, де зазначена інформація циркулює, а також відкритої інформації, важливої для особи та суспільства, якщо ця інформація циркулює в органах державної влади та органах місцевого самоврядування, Національній академії наук, Збройних Силах, інших військових формуваннях, органах внутрішніх справ, на державних підприємствах, у державних установах та організаціях; - виконання на власний розсуд суб'єктами інформаційних відносин вимог щодо технічного захисту конфіденційної інформації, що належить державі, та відкритої інформації, важливої для особи та суспільства, якщо остання циркулює поза межами державних органів, підприємств, установ і організацій; - покладення відповідальності за формування та реалізацію державної політики у сфері технічного захисту інформації на спеціально уповноважений центральний орган виконавчої влади; - ієрархічність побудови організаційних структур системи захисту інформації та керівництво їх діяльністю у межах повноважень, визначених нормативно-правовими актами; - методичне керівництво спеціально уповноваженим центральним органом виконавчої влади у сфері захисту інформації діяльністю організаційних структур системи технічного захисту інформації; - скоординованість дій та розмежування сфер діяльності організаційних структур системи технічного захисту інформації з іншими системами захисту інформації та системами забезпечення інформаційної безпеки, як складової національної безпеки; - фінансова забезпеченість системи захисту інформації за рахунок Державного бюджету України, бюджету Автономної Республіки Крим, місцевих

бюджетів та інших джерел. З аналізу нормативно-правових актів України витікає, що державна політика у сфері захисту інформації визначається пріоритетністю національних інтересів, має на меті унеможливлення реалізації загроз для інформації та здійснюється шляхом виконання положень, зазначених у законодавстві та положень Концепції технічного захисту інформації, а також програм розвитку захисту інформації та окремих проектів.

1.2.Комп'ютер очима хакера

У діловому світі нарешті визнали важливість вирішення проблеми захисту комп'ютерних даних. Гучні процеси, пов'язані з проникненням зловмисників в корпоративні комп'ютерні системи, особливо поділи Левіна, назване Інтерполом найсерйознішим транснаціональним мережевим комп'ютерним злочином, внаслідок якого американський Сіті-банк втратив 400 тис. доларів, привернули пильну увагу не тільки фахівців в області комп'ютерної обробки даних, але і директорів компаній. Останні нехай із запізненням, але все-таки зрозуміли, що з пуском в експлуатацію кожної нової комп'ютерної системи, що має вихід у глобальну комп'ютерну мережу Internet, вони ризикують розкрити перед зловмисниками всіх мастей (професійними зломщиками і грабіжниками, скривдженими підлеглими або нічим не погребують конкурентами) вікно, через яке ті можуть безперешкодно проникати у святая святих компанії і наносити істотної матеріальної шкоди. В результаті як необізнаність керівників, так і бюджетні обмеження тепер не є основними перешкодами на шляху впровадження заходів захисту інформації в комп'ютерних системах, а головну роль грає вибір конкретних інструментів і рішень.

З'ясувалося, що створення добре захищеної комп'ютерної системи неможливо без ретельного аналізу потенційних загроз для її безпеки. Фахівці склали перелік дій, які потрібно зробити в кожному конкретному випадку, щоб представляти сценарії можливих нападів на комп'ютерну систему. Цей перелік включав:

- визначення цінності інформації, що зберігається в комп'ютерній системі:
- оцінку тимчасових і фінансових витрат, які може дозволити собі зловмисник для подолання механізмів захисту комп'ютерної системи;
- ймовірну модель поведінки зловмисника при атаці на комп'ютерну систему;
- оцінку тимчасових і фінансових витрат, необхідних для організації адекватного захисту комп'ютерної системи.

Таким чином, при проведенні аналізу потенційних загроз безпеки комп'ютерної системи експерт ставив себе на місце зловмисника, який намагається проникнути в цю систему. А для цього йому необхідно було зрозуміти, що являє собою зловмисник, від якого потрібно захищатися. І в першу чергу потрібно було якомога точніше відповісти на наступні питання:

• наскільки високий рівень професійної підготовки зловмисника;

• якою інформацією про атакуєму комп'ютерну систему він володіє;

• як зловмисник здійснює доступ до цієї системи;

• яким способом атаки він скористається з найбільшою ймовірністю.

Проте експертам, яким доводилося ставити себе на місце зловмисника, щоб відповісти на перелічені вище запитання, дуже не сподобалося іменуватися зловмисниками. По-перше, це слово - довге й незграбне, а по-друге, воно не зовсім адекватно відображає суть розв'язуваної задачі, яка полягає не в знаходженні проломів у захисті комп'ютерної системи, а в їх ліквідації. Тому вони взяли на озброєння інший термін, який точніше відповідає покладеним на них місії, і стали розробляти сценарії поведінки так званих хакерів.

1.3.Хто такі хакери

Хоча останнім часом термін хакер можна досить часто зустріти на сторінках комп'ютерної преси, люди до цих пір не склалося єдиної думки про те, кого саме слід іменувати хакером. Часто хакером називають будь-якого висококласного фахівця в галузі обчислювальної техніки і всього, що з нею пов'язано. Однак серед журналістів, які зачіпають тему хакерства, є серйозні розбіжності щодо того, як хакери застосовують свої унікальні знання на практиці.

Одні пропонують називати хакером лише того, хто намагається зламати захист комп'ютерних систем, щоб потім видати обгрунтовані рекомендації щодо поліпшення їх захисних механізмів, інші - іменувати хакером висококваліфікованого фахівця, який зламує комп'ютери виключно в злочинних цілях. Щоб не вплутуватися в полеміку, яка в основному стосується морально-етичної сторони діяльності хакерів, для початку назвемо хакером будь-якої людини, що прагне обійти захист комп'ютерної системи, незалежно від того, переслідуються за законом його дії. При цьому основною метою хакера є отримання додаткових привілеїв і прав доступу до комп'ютерної системи. По відношенню до атакуємої їм комп'ютерній системі хакер може бути:

• сторонньою особою, не має жодних легальних привілеїв і прав доступу;

• користувачем комп'ютерної системи, що володіє обмеженими привілеями та правами доступу.

Статистика комп'ютерних злочинів свідчить про те, що рівень професійної підготовки хакера варіюється в дуже широких межах. Хакером може стати навіть школяр, випадково виявив програму злому на одному із спеціалізованих хакерських серверів та мережі Internet. У той же час

відзначено і поява справжніх хакерських банд, ватажками яких є комп'ютерні фахівці найвищої кваліфікації.

Надалі під хакером буде розумітися тільки високо кваліфікований фахівець, оскільки саме його дії являють найбільшу загрозу безпеці комп'ютерних систем. Для такого хакера характерні наступні риси та особливості поведінки:

• він завжди в курсі останніх новинок у галузі комп'ютерної техніки пристроїв зв'язку та програмних засобів;

• перед тим як атакувати комп'ютерну систему, він всіма доступними способами намагається зібрати максимум інформації про цю систему, включаючи дані про використовуваний у ній програмному забезпеченні та її адміністраторах;

• добуваючи потрібну йому інформацію, він не потребує агентурними та оперативно-технічними методами (наприклад, встановлюючи підслуховуючі пристрої в місцях, часто відвідуваних обслуговуючим персоналом комп'ютерних систем, які він має намір зламати);

• перед спробою злому комп'ютерної системи він випробує методи, які планує застосувати для атаки на цю систему, на заздалегідь підготовленій моделі, що має ті ж засоби забезпечення безпеки, що і атакують система;

• сама атака комп'ютерної системи здійснюється по можливості швидко, щоб її адміністратори не змогли зафіксувати факт вчинення атаки і не встигли вжити заходів для відбиття атаки і для виявлення особи і місцезнаходження атакуючого;

• хакер не користується надто витонченими методами злому запиті комп'ютерної системи, оскільки чим складніше алгоритм атаки, тим імовірніше виникнення помилок і збоїв при його реалізації;

• щоб мінімізувати час, необхідний для злому, і кількість можливих помилок, хакер зазвичай атакує комп'ютерну систему за допомогою заздалегідь написаних програм, а не вручну, набираючи необхідні команди на клавіатурі комп'ютера;

• хакер ніколи не діє під власним ім'ям і ретельно приховує свій мережевий адресу, а про всяк випадок у нього є ретельно продуманий план замести сліди або залишити помилковий слід (наприклад. одночасно ведучи невмілу і свідомо приречену на провал атаку, завдяки чому журнал аудиту атакується комп'ютерної системи виявляється забитий повністю повідомленнями про події, що ускладнюють для системного адміністратора з'ясування характеру дійсної атаки і вжиття заходів, щоб не допустити її в майбутньому);

• хакери широко застосовують програмні закладки, які самознищується або при їх виявленні, або після закінчення фіксованого періоду часу.

1.4.Методи злому комп'ютерних систем

У загальному випадку програмне забезпечення будь-якої універсальної комп'ютерної системи складається з трьох основних

компонентів: операційної системи, мережевого програмного забезпечення (МПЗ) та системи управління базами даних (СУБД). Тому всі спроби злому захисту комп'ютерних систем можна розділити на три групи:

• атаки на рівні операційної системи;
• атаки на рівні мережного програмного забезпечення;
• атаки на рівні систем управління базами даних.

Атаки на рівні систем управління базами даних

Захист СУБД є однією з найпростіших завдань. Це пов'язано з тим, що СУБД мають строго певну внутрішню структуру, і операції над елементами СУБД задані досить чітко. Є чотири основні дії - пошук, вставка, видалення і заміна елементу. Інші операції є допоміжними і застосовуються досить рідко. Наявність строгої структури і чітко визначених операцій спрощує рішення задачі захисту СУБД. У більшості випадків хакери вважають за краще зламувати захист комп'ютерної системи на рівні операційної системи і отримувати доступ до файлів СУБД за допомогою засобів операційної системи. Однак у випадку, якщо використовується СУБД, яка не має достатньо надійних захисних механізмів, чи погано протестована версія СУБД, що містить помилки, або якщо при визначенні політики безпеки адміністратором СУБД були допущені помилки, то стає цілком ймовірним подолання хакером захисту, що реалізується на рівні СУБД.

Крім того, є два специфічних сценарію атаки на СУБД, для захисту від яких потрібно застосовувати спеціальні методи. У першому випадку результати арифметичних операцій над числовими полями СУБД округляються в меншу сторону, а різниця підсумовується в деякій іншого запису СУБД (як правило, цей запис містить особистий рахунок хакера в банку, а округляемое числові поля відносяться до рахунків інших клієнтів банку). У другому випадку хакер отримує доступ до полів записів СУБД, для яких доступною є тільки статистична інформація. Ідея хакерської атаки на СУБД - так хитро сформулювати запит, щоб безліч записів, для якого збирається статистика, складалося тільки з одного запису.

Атаки на рівні операційної системи

Захищати операційну систему, на відміну від СУБД, набагато складніше. Справа в тому, що внутрішня структура сучасних операційних систем надзвичайно складна, і тому дотримання адекватної політики безпеки є значно більш важким завданням. Серед людей недосвідчених існує думка, що найефективніші атаки на операційні системи можуть бути організовані тільки за допомогою складних засобів, заснованих на новітніх досягненнях науки і техніки, а хакер повинен бути програмістом високої кваліфікації. Це не зовсім так.

Ніхто не сперечається з тим, що користувачу слід бути в курсі всіх новинок в області комп'ютерної техніки. Та й висока кваліфікація - зовсім не зайве. Проте мистецтво хакера полягає зовсім не в тому, щоб зламувати будь-яку саму "крутий" комп'ютерний захист. Потрібно просто зуміти знайти

слабке місце в конкретній системі захисту. При цьому найпростіші методи злому виявляються нітрохи не гірше самих витончених, оскільки чим простіше алгоритм атаки, тим більше вірогідність її завершення без помилок і збоїв, особливо якщо можливості попереднього тестування цього алгоритму в умовах, наближених до "бойових", дуже обмежені

Успіх реалізації того чи іншого алгоритму хакерської атаки на практиці в значній мірі залежить від архітектури і конфігурації конкретної операційної системи, що є об'єктом цієї атаки. Проте є атаки, яким може бути піддана практично будь-яка операційна система:

• крадіжка пароля;

• підглядання за користувачем, коли той вводить пароль, що дає право на роботу з операційною системою (навіть якщо під час введення пароль не висвічується на екрані дисплея, хакер може легко у шип, пароль, просто стежачи за переміщенням пальців користувача по клавіатурі);

• отримання пароля з файлу, в якому цей пароль був збережений користувачем, не охочим утрудняти себе введенням пароля при підключенні до мережі (як правило, такий пароль зберігається у файлі в незашифрованому вигляді);

• пошук пароля, який користувачі, щоб не забути, записують па календарях, у записних книжках або на зворотному боці комп'ютерних клавіатур (особливо часто подібна ситуація зустрічається, якщо адміністратори примушують користувачів застосовувати важко запам'ятовуються паролі);

• крадіжка зовнішнього носія парольної інформації (дискети або електронного ключа, на яких зберігається пароль користувача, призначений для входу в операційну систему);

• повний перебір всіх можливих варіантів пароля;

• підбір пароля по частоті символів і біграмм, за допомогою словників найчастіше вживаних паролів, із залученням знань про конкретного користувача - його імені, прізвища, номери телефону, дати народження і т. д., з використанням відомостей про існування еквівалентних паролів, при цьому з кожного класу випробовується всього один пароль, що може значно скоротити час перебору;

• сканування жорстких дисків комп'ютера (хакер послідовно намагається звернутися до кожного файлу, що зберігається на жорстких дисках комп'ютерної системи; якщо обсяг дискового простору достатньо великий, можна бути цілком впевненим, що при описі доступу до файлів і каталогів адміністратор допустив хоча б одну помилку, в результаті чого всі такі каталоги і файли будуть прочитані хакером; для приховування слідів хакер може організувати цю атаку під чужим ім'ям: наприклад, під ім'ям користувача, пароль якого відомий хакеру);

• збірка "сміття" (якщо засоби операційної системи дозволяють відновлювати раніше видалені об'єкти, хакер може скористатися цією

можливістю, щоб отримати доступ до об'єктів, вилученим іншими користувачами: наприклад, переглянувши вміст їх "сміттєвих" кошиків);

• перевищення повноважень (використовуючи помилки в програмному забезпеченні або в адмініструванні операційної системи, хакер отримує повноваження, що перевищують повноваження, надані йому відповідно до чинної політики безпеки);

• запуск програми від імені користувача, що має необхідні повноваження, або як системної програми (драйвера, сервісу, демона і т. д.);

• підміна динамічно завантажується бібліотеку до системними програмами, або зміна змінних середовища, що описують шлях до таких бібліотеках;

• модифікація коду або даних підсистеми захисту самої операційної системи;

• відмову в обслуговуванні (метою цієї атаки є частковий або повний вивід з ладу операційної системи);

• захоплення ресурсів (хакерська програма проводить захоплення всіх наявних в операційній системі ресурсів, а потім входить в нескінченний цикл);

• бомбардування запитами (хакерська програма постійно направляє операційній системі запити, реакція на які вимагає залучення значних ресурсів комп'ютера);

• використання помилок в програмному забезпеченні або адмініструванні.

Якщо в програмному забезпеченні комп'ютерної системи немає помилок і її адміністратор суворо дотримується політики безпеки, рекомендовану розробниками операційної системи, то атаки всіх перерахованих піки, малоефективні. Додаткові заходи, які повинні бути зроблені для підвищення рівня безпеки, в значній мірі залежать від конкретної операційної системи, під управлінням якої працюємо дана комп'ютерна система. Тим не менш, доводиться визнати, що незалежно від вжитих заходів повністю усунути загрозу злому комп'ютерної системи на рівні операційної системи неможливо. Тому політика забезпечення безпеки повинна проводитися так, щоб, навіть подолавши захист, створювану засобами операційної системи, хакер не зміг завдати серйозного збитку.

Атаки на рівні мережного програмного забезпечення

СПО є найбільш вразливим, тому що канал зв'язку, по якому передаються повідомлення, найчастіше не захищений, і кожен, хто може мати доступ до цього каналу, відповідно, може перехоплювати повідомлення і відправляти свої власні. Тому на рівні МПЗ можливі наступні хакерські атаки:

• прослуховування сегменту локальної мережі (в межах одного й того ж сегменту локальної мережі приєднаному до нього комп'ютер в змозі приймати повідомлення, адресовані іншим комп'ютерам сегменту, а отже,

якщо комп'ютер хакера приєднаний до деякого сегменту локальної мережі, то йому стає доступний весь інформаційний обмін між комп'ютерами цього сегмента);

• перехоплення повідомлень на маршрутизаторі (якщо хакер має привілейований доступ до мережного маршрутизатора, то він отримує можливість перехоплювати всі повідомлення, що проходять через цей маршрутизатор, і хоча тотальний перехоплення неможливий через занадто великого обсягу, надзвичайно привабливим для хакера є вибіркове перехоплення повідомлень, що містять паролі користувачів і їх електронну пошту);

• створення помилкового маршрутизатора (шляхом відправки в мережу повідомлень спеціального виду хакер добивається, щоб його комп'ютер став маршрутизатором мережі, після чого отримує доступ до всіх хто проходить через його повідомлення);

• нав'язування повідомлень (відправляючи в мережу повідомлення з помилковим зворотним мережною адресою, хакер перемикає на свій комп'ютер вже встановлені мережні з'єднання і в результаті отримує права користувачів, чиї з'єднання обманним шляхом були переключені на комп'ютер хакера);

• відмову в обслуговуванні (хакер відправляє в мережу повідомлення спеціальною виду, після чого одна або декілька комп'ютерних систем, підключених до мережі, повністю або частково виходять з ладу).

Оскільки хакерські атаки на рівні МПЗ спровоковані відкритістю мережних з'єднань, розумно припустити, що дли відображення цих атак необхідно максимально захистити канали зв'язку і тим самим ускладнити обмін інформацією з мережі для тих, хто не є легальним користувачем. Нижче перераховані деякі способи такого захисту:

• максимальне обмеження розмірів комп'ютерної мережі (чим більше мережа, тим важче її захистити);

• ізоляція мережі від зовнішнього світу (по можливості слід обмежувати фізичний доступ до комп'ютерної мережі ззовні, щоб зменшити вірогідність несанкціонованого підключення хакера);

• шифрування мережних повідомлень (тим самим можна усунути загрозу перехоплення повідомлень, правда, за рахунок зниження продуктивності СПО і зростання накладних витрат);

• електронний цифровий підпис мережних повідомлень (якщо всі повідомлення, передані по комп'ютерній мережі, забезпечуються електронним цифровим підписом, і при цьому непідписані повідомлення ігноруються, то можна забути про загрозу нав'язування повідомлень і про більшість загроз, пов'язаних з відмовою в обслуговуванні);

• використання брандмауерів (брандмауер є допоміжним засобом захисту, застосовуваним тільки в тому випадку, якщо комп'ютерну мережу не можна ізолювати від інших мереж, оскільки брандмауер досить часто не здатний відрізнити потенційно небезпечне мережне повідомлення від

абсолютно нешкідливого, і в результаті типовою є ситуація, коли брандмауер не тільки не захищає мережу від хакерських атак, а й навіть перешкоджає її нормальному функціонуванню).

1.5.Захист системи від злому

Перераховані вище методи хакерської атаки на комп'ютерну систему є найбільш типовими і описані в загальній формі. Найпоширеніші з цих методів будуть розглянуті нижче більш детально, оскільки їх застосування в конкретних випадках має свої особливості, які вимагають застосування додаткових захисних заходів. А поки для узагальненої моделі злому комп'ютерних систем можна сформулювати універсальні правила, яких слід дотримуватися, щоб звести ризик до мінімуму.

• Не відставайте від хакерів: будьте завжди в курсі останніх розробок в галузі комп'ютерної безпеки. Оформіть передплату на кілька спеціалізованих журналів, в яких докладно висвітлюються питання захисту комп'ютерних систем від злому. Регулярно переглядав матеріали, що поміщаються на хакерських серверах Internet (наприклад. astalavista.box.sk).

• Керуйтеся принципом розумної достатності: не прагнете побудувати абсолютно надійний захист. Адже чим потужніший зашита, тим більше ресурсів комп'ютерної системи вона споживає і тим важче використовувати її.

• Зберігайте в секреті інформацію про принципи дії захисних механізмів комп'ютерної системи. Чим менше хакеру відомо про ці принципи, тим важче буде для нього організувати успішну атаку.

• Постарайтеся максимально обмежити розміри захищається комп'ютерної мережі та без крайньої необхідності не допускайте її підключення до Internet.

• Перед тим як вкласти кошти в купівлю нового програмного забезпечення, пошукайте інформацію про нього, наявну на хакерських серверах Internet.

• Розміщуйте сервери в охоронюваних приміщеннях. Не підключайте до них клавіатуру і дисплеї, щоб доступ до цих серверів здійснювався тільки через мережу.

• Абсолютно всі повідомлення, що передаються по незахищених каналах зв'язку, повинні шифруватися і забезпечуватися цифровим підписом.

• Якщо захищається комп'ютерна мережа має з'єднання з незахищеною мережею, то всі повідомлення, що відправляються в цю мережу або прийняті з неї повинні проходити через брандмауер, а також шифруватися і забезпечуватися цифровим підписом.

• Не нехтуйте можливостями, які надає аудит. Інтервал між сеансами перегляду журналу аудиту не повинен перевищувати однієї доби.

• Якщо виявиться, що кількість подій, приміщенні в журнал аудиту, надзвичайно велика, вивчіть уважно всі нові записи. оскільки не виключено, що комп'ютерна система піддалася атаці хакера, який намагається замести сліди свого нападу, зафіксовані в журналі аудиту.

• Регулярно проводите перевірку цілісності програмного забезпечення комп'ютерної системи. Перевірки її на наявність програмних закладок.

• Реєструйте всі зміни політики безпеки в звичайному паперовому журналі. Регулярно звіряйте політику безпеки із зареєстрованою в цьому журналі. Це допоможе виявити присутність програмної закладки, якщо вона була впроваджена хакером в комп'ютерну систему.

• Користуйтеся захищеними операційними системами.

• Створіть кілька пасток для хакерів (наприклад, заведіть на диску файл з привабливим ім'ям, прочитати який неможливо за допомогою звичайних засобів, і якщо буде зафіксовано успішне звернення до цього файлу, значить в захищається комп'ютерну систему була впроваджена програмна закладка).

• Регулярно тестуйте комп'ютерну систему за допомогою спеціальних програм, призначених для визначення ступеня її захищеності від хакерських атак.

2. Програми-шпигуни

2.1.Программние закладки

Сучасна концепція створення комп'ютерних систем передбачає використання програмних засобів різного призначення в єдиному комплексі. Наприклад, типова система автоматизованого документообігу складається з операційного середовища, програмних засобів управління базами даних, телекомунікаційних програм, текстових редакторів, антивірусних моніторів, засобів для криптографічного захисту даних, а також засобів аутентифікації та ідентифікації користувачів. Головною умовою правильного функціонування такої комп'ютерної системи є забезпечення захисту від втручання в процес обробки інформації тих програм, присутність яких в комп'ютерній системі не обов'язково. Серед подібних програм, в першу чергу, слід згадати комп'ютерні віруси. Проте є шкідливі програми ще одного класу. Від них, як і від вірусів, слід з особливою ретельністю очищати свої комп'ютерні системи. Це так звані програмні закладки, які можуть виконувати хоча б одну з таких дій:

• вносити довільні спотворення в коди програм, що знаходяться і оперативної пам'яті комп'ютера (програмна закладка першого типу);

• переносити фрагменти інформації з одних областей оперативної або зовнішньої пам'яті комп'ютера в інші (програмна закладка другого типу);

• спотворювати виведену на зовнішні комп'ютерні пристрої або в канал зв'язку інформацію, отриману в результаті роботи інших програм (програмна закладка третього типу).

Програмні закладки можна класифікувати і за методом їх впровадження в комп'ютерну систему:

• програмно-апаратні закладки, асоційовані з апаратними засобами комп'ютера (їх середовищем існування, як правило, є BIOS - набір програм, записаних у вигляді машинного коду в постійному запам'ятовуючому пристрої - ПЗУ);

• завантажувальні закладки, асоційовані з програмами початкового завантаження, які розташовуються в завантажувальних секторах (з цих секторів у процесі виконання початкового завантаження комп'ютер зчитує програму, що бере на себе управління для подальшого завантаження самої операційної системи);

• драйверні закладки, асоційовані з драйверами (файлами, і яких міститься інформація, необхідна операційній системі для управління підключеними до комп'ютера периферійними пристроями);

• прикладні закладки, асоційовані з прикладним програмним забезпеченням загального призначення (текстові редактори, утиліти, антивірусні монітори і програмні оболонки);

• виконувані закладки, асоційовані з виконуваними програмними модулями, що містять код цієї закладки (найчастіше ці модулі є пакетні файли, тобто файли, які складаються з команд операційної системи, виконуваних одна за однією, як якщо б їх набирали на клавіатурі комп'ютера);

• закладки-імітатори, інтерфейс яких збігається з інтерфейсом деяких службових програм, що вимагають введення конфіденційної інформації (паролів, криптографічних ключів, номерів кредитних карток);

• замасковані закладки, які маскуються під програмні засоби оптимізації роботи комп'ютера (файлові архіватори, дискові дефрагментатори) або під програми ігрового і розважального призначення.

Щоб програмна закладка могла зробити будь-які дії по відношенню до інших програм чи по відношенню до даних, процесор повинен приступити до виконання команд, що входять до складу коду програмної закладки. Це можливо тільки при одночасному дотриманні наступних умов:

• програмна закладка повинна потрапити в оперативну пам'ять комп'ютера (якщо закладка відноситься до першого типу, то вона повинна бути завантажена до початку роботи іншої програми, яка є метою впливу закладку, або під час роботи цієї програми);

• робота закладки, що знаходиться в оперативній пам'яті, починається при виконанні ряду умов, які називаються активизирующими.

Цікаво, що іноді сам користувач провокується на запуск виконуваного файлу, що містить код програмної закладки. Відомий такий випадок. Серед користувачів вільно поширювався набір з файлів, що

архівуються. Для вилучення файлів з нього вимагалося викликати спеціальну утиліту, яка, як правило, є майже у кожного користувача і запускається після вказівки її імені в командному рядку. Однак мало хто з користувачів помічав, що в отриманому наборі файлів вже була програма з таким же ім'ям і. Що запускати саме вона. Крім розархівування файлів, ця програмна закладка додатково виробляла ряд дій негативного характеру.

З урахуванням зауваження про те, що програмна закладка повинна бути обов'язково завантажена в оперативну пам'ять комп'ютера, можна виділити резидентні закладки (вони знаходяться в оперативній пам'яті постійно, починаючи з деякого моменту і до закінчення сеансу роботи комп'ютера, т. з. До його перезавантаження або до вимикання живлення) та нерезидентні (такі закладки потрапляють в оперативну пам'ять комп'ютера аналогічно резидентним, проте, на відміну від останніх, вивантажуються після закінчення деякого часу або при виконанні особливих умов).

Існують три основні групи деструктивних дій, які можуть здійснюватися програмними закладками:

• копіювання інформації користувача комп'ютерної системи (паролів, криптографічних ключів, кодів доступу, конфіденційних електронних документів), що знаходиться в оперативній або зовнішньої пам'яті цієї системи або в пам'яті іншої комп'ютерної системи, підключеної до неї через локальну або глобальну комп'ютерну мережу;

• зміна алгоритмів функціонування системних, прикладних м службових програм (наприклад, внесення змін до програми розмежування доступу може призвести до того, що вона дозволить вхід в систему всім без виключення користувачам незалежно від правильності введеного пароля);

• нав'язування певних режимів роботи (наприклад, блокування запису на диск при видаленні інформації, при цьому інформація, яку потрібно видалити, не знищується і може бути згодом скопійована хакером).

У всіх програмних закладок (незалежно від способу їх впровадження в комп'ютерну систему, терміну їх перебування в оперативній пам'яті і призначення) є одна важлива спільна риса: вони обов'язково виконують операцію запису в оперативну або зовнішню пам'ять системи. При відсутності цієї операції ніякого негативного впливу програмна закладка надати не може. Ясно, що для цілеспрямованого впливу вона повинна виконувати і операцію читання, інакше в ній може бути реалізована тільки функція руйнування (наприклад, видалення або заміна інформації в певних секторах жорсткого диска).

2.2.Моделі впливу програмних закладок на комп'ютери

Перехоплення

У моделі перехоплення програмна закладка впроваджується в ПЗУ. системне або прикладне програмне забезпечення та зберігає всю або вибрану інформацію, що вводиться з зовнішніх пристроїв комп'ютерної системи або виводиться на ці пристрої, у прихованій області пам'яті локальної або

віддаленої комп'ютерної системи. Об'єктом збереження, наприклад, можуть служити символи, введені з клавіатури (всі повторювані два рази послідовності символів), або електронні документи, роздруковуються на принтері.

Дана модель може бути двоступеневою. На першому етапі зберігаються тільки, наприклад, імена або початку файлів. На другому накопичені дані аналізуються зловмисником з метою прийняття рішення про конкретних об'єктах подальшої атаки.

Модель типу "перехоплення" може бути ефективно використана при атаці на захищену операційну систему Windows NT. Після старту Windows NT на екрані комп'ютерної системи з'являється запрошення натиснути клавіші <Ctrl> + <Alt> + <Del>. Після їх натиснення завантажується динамічна бібліотека MSGINA.DLL, що здійснює прийом вводиться пароля та виконання процедури його перевірки (аутентифікації). Опис усіх функцій цієї бібліотеки можна знайти у файлі Winwlx.h. Також існує простий механізм заміни вихідної бібліотеки MSGINA.DLL на налаштовувану (для цього необхідно просто додати спеціальну рядок до реєстру операційної системи Windows NT і вказати місце розташування користувача бібліотеки). У результаті зловмисник може модифікувати процедуру контролю за доступом до комп'ютерної системи, що працює під управлінням Windows NT.

Спотворення

У моделі спотворення програмна закладка змінює інформацію, яка записується в пам'ять комп'ютерної системи в результаті роботи програм, або придушує / ініціює виникнення помилкових ситуацій у комп'ютерній системі.

Можна виділити статичне і динамічне спотворення. Статичний спотворення відбувається всього один раз. При цьому модифікуються параметри програмного середовища комп'ютерної системи, щоб згодом у ній виконувалися потрібні зловмисникові дії. До статичному спотворення відноситься, наприклад, внесення змін в файл AUTOEXEC.BAT операційної системи Windows 95/98, які призводять до запуску заданої програми, перш ніж будуть запущені всі інші, перелічені в цьому файлі.

Фахівцям російського Федерального агентства урядового зв'язку та інформації (ФАПСИ) вдалося виявити при аналізі однієї з вітчизняних систем цифрового підпису цікаве статистичне спотворення Зловмисник (співробітник відділу інформатизації фінансової організації, в якій була впроваджена ця система) виправив у виконуваному EXE-модулі програми перевірки правильності цифрового підпису символьну рядок "ПІДПИС некоректно" на символьну рядок "ПІДПИС коректно". В результаті взагалі перестали фіксуватися документи з невірними цифровими підписами, і, отже, в електронні документи стало можна вносити довільні зміни вже після їх підписання електронним цифровим підписом.

Динамічне спотворення полягає у зміні будь-яких параметром системних або прикладних процесів за допомогою заздалегідь активізованих закладок. Динамічне спотворення можна умовно розділити так: спотворення

на вході (коли на обробку потрапляє вже спотворений документ) та спотворення на виході (коли спотворюється інформація, яка відображається для сприйняття людиною, або призначена для роботи інших програм).

Практика застосування цифрового підпису в системах автоматизованого документообігу показала, що саме програмна реалізація цифрового підпису особливо схильна до впливу програмних закладок типу "динамічний спотворення", які дозволяють здійснювати проводки фальшивих фінансових документів і втручатися в процес вирішення спорів за фактами неправомірного застосування цифрового підпису. Наприклад, в одній із програмних реалізацій широко відомої криптосистеми PGP електронний документ, під яким вимагалося поставити цифровий підпис, зчитується блоками по 512 байт, причому процес зчитування вважався завершеним, якщо в прочитаному блоці дані займали менше 512 байт. Робота однієї програмної закладки, виявленої фахівцями ФАПСИ, грунтувалася на нав'язуванні довжини файлу. Ця закладка дозволяла зчитувати тільки перші 512 байт документа, і в результаті цифровий підпис визначалася на основі лише цих 512 байт. Така ж схема діяла і при перевірці поставленої під документом цифрового підпису. Отже, що залишилася частина цього документа могла бути довільним чином викривлена, і цифровий підпис під ним продовжувала залишатися "коректною".

Існують 4 основних способи впливу програмних закладок на цифровий підпис:

• спотворення вхідної інформації (змінюється надходить на підпис електронний документ);

• спотворення результату перевірки істинності цифрового підпису (незалежно від результатів роботи програми цифровий підпис оголошується справжньої);

• нав'язування довжини електронного документа (програмі цифрового підпису пред'являється документ меншої довжини, ніж насправді, і в результаті цифрові Повний ставиться тільки під частиною вихідною документа);

• спотворення програми цифрового підпису (вносяться зміни в виконуваний код програми з метою модифікації реалізованого алгоритму).

В рамках моделі "спотворення" також реалізуються програмні закладки. дія яких грунтується на ініціювання або придушенні сигналу про виникнення помилкових ситуацій в комп'ютерній системі, тобто тих, які призводять до відмінного від нормального завершення виконуваної програми (встановленого відповідною документацією).

Для ініціювання статичної помилки на пристроях зберігання інформації створюється область, при зверненні до якої (читання, запис, форматування і т. п.) виникає помилка, що може утруднити чи блокувати деякі небажані для зловмисника дії системних пли прикладних програм (наприклад, не дозволяти здійснювати коректно знищити конфіденційну інформацію на жорсткому диску).

При ініціюванні динамічної помилки для деякої операції генерується помилкова помилка з числа тих помилок, які можуть виникати при виконанні даної операції. Наприклад, для блокування прийому або передачі інформації в комп'ютерній системі може постійно ініціюватися помилкова ситуація "МОДЕМ зайнятий". Або при прочитанні першого блоку інформації довжиною 512 байт може встановлюватися відповідний прапорець для того, щоб не допустити прочитання другого і наступних блоків і в підсумку підробити цифровий підпис під документом.

Щоб маскувати помилкові ситуації, зловмисники зазвичай використовують придушення статичної або динамічної помилки. Метою такого придушення часто є прагнення блокувати нормальне функціонування комп'ютерної системи або бажання змусити її неправильно працювати. Надзвичайно важливо, щоб комп'ютерна система адекватно реагувала на виникнення всіх без винятку помилкових ситуацій, оскільки відсутність належної реакції на будь-яку помилку еквівалентно її придушення і може бути використано зловмисником. Відомий випадок успішної атаки пари аргентинських літаків-торпедоносців на англійський есмінець "Шеффілд", що закінчився нанесенням серйозних пошкоджень цього корабля. Через помилки в програмному забезпеченні встановлена на ньому система протиповітряної оборони не змогла вибрати мету, яку належало збивати першої, оскільки атакуючі літаки летіли дуже близько один від одного.

Різновидом перекручення є також модель типу троянський кінь. У цьому випадку програмна закладка вбудовується в постійно використовуване програмне забезпечення та по деякому активізує події викликає виникнення збійної ситуації в комп'ютерній системі. Тим самим досягаються відразу дві мети: паралізується її нормальне функціонування, а зловмисник, отримавши доступ до комп'ютерної системи для усунення неполадок, зможе, наприклад, витягти з неї інформацію, перехоплену іншими програмними закладками. Як активизирующего події зазвичай використовується наступ певного момент;: часу, сигнал з каналу модемного зв'язку або стан деяких лічильників (наприклад, лічильника кількості запусків програми).

Прибирання сміття

Як відомо, при зберіганні комп'ютерних даних на зовнішніх носіях прямого доступу виділяється кілька рівнів ієрархії: сектора, кластери і файли. Сектора є одиницями зберігання інформації на апаратному рівні. Кластери складаються з одного або кількох поспіль секторів. Файл - це безліч кластерів, пов'язаних з певним законом.

Робота з конфіденційними електронними документами зазвичай зводиться до послідовності наступних маніпуляцій з файлами:

- створення;
- зберігання;
- корекція;
- знищення.

Для захисту конфіденційної інформації зазвичай використовується шифрування. Основна загроза виходить зовсім не від використання нестійких алгоритмів шифрування і "поганих" криптографічних ключів (як це може здатися на перший погляд), а від звичайних текстових редакторів і баз даних, що застосовуються для створення та корекції конфіденційних документів!

Справа в тому, що подібні програмні засоби, як правило, у процесі функціонування створюють в оперативній або зовнішньої пам'яті комп'ютерної системи тимчасові копії документів, з якими вони працюють. Природно, всі ці тимчасові файли випадають з поля зору будь-яких програм шифрування і можуть бути використані зловмисником для того, щоб скласти уявлення про зміст зберігаються в зашифрованому вигляді конфіденційних документів.

Важливо пам'ятати і про те, що при записі відредагованій інформації меншого обсягу в той же файл, де зберігалася початкова інформація до початку сеансу її редагування, утворюються так звані "хвостові" кластери, в яких ця вихідна інформація повністю зберігається. І тоді "хвостові" кластери не тільки не піддаються впливу програм шифрування, але і залишаються незачепленими навіть засобами гарантованого стирання інформації. Звичайно, рано чи пізно інформація з "хвостових" кластерів затирається даними з інших файлів, однак за оцінками фахівців ФАПСИ з "хвостових" кластерів через добу можна витягти до 85%, а через десять діб - до 25-40% вихідної інформації.

Користувачам необхідно мати на увазі й те, що команда видалення файлу (DEL) операційної системи DOS2 не змінює змісту файлу, і воно може бути в будь-який момент відновлено, якщо поверх нього ще не був записаний інший файл. Поширені кошти гарантованого стирання файлів попередньо записують на його місце константи або випадкові числа і тільки після цього видаляють файл стандартними засобами DOS. Однак навіть такі потужні засоби виявляються безсилими проти програмних закладок, які націлені на те, щоб збільшити кількість залишених у вигляді "сміття" фрагментів конфіденційної інформації. Наприклад, програмна закладка може ініціювати статичну помилку, помітивши один або кілька кластерів з ланцюжка, що входить в файл, міткою "збійні". В результаті при видаленні файла засобами операційної системи або засобами гарантованого знищення та його частина, яка розміщена в збійних кластерах, залишиться недоторканою і згодом може бути відновлена за допомогою стандартних утиліт.

Спостереження і компрометація

Крім перерахованих, існують і інші моделі впливу програмних закладок на комп'ютери. Зокрема, при використанні моделі типу спостереження програмна закладка вбудовується в мережеве або телекомунікаційне програмне забезпечення. Користуючись тим, що подібне програмне забезпечення завжди знаходиться в стані активності, запроваджена в нього програмна закладка може стежити за всіма процесами обробки інформації в комп'ютерній системі, а також здійснювати установку і

видалення інших програмних закладок. Модель типу компрометація дозволяє отримувати доступ до інформації, перехопленої іншими програмними закладками. Наприклад, ініціюється постійне звернення до такої інформації, що приводить до зростання співвідношення сигнал / шум. А це, в свою чергу, значно полегшує перехоплення побічних випромінювань даної комп'ютерної системи і дозволяє ефективно виділяти сигнали, згенеровані закладкою типу "компрометація", із загального фону випромінювання, що виходить від устаткування.

2.3.Захист від програмних закладок

Завдання захисту від програмних закладок може розглядатися в трьох принципово різних варіантах:

• не допустити впровадження програмної закладки в комп'ютерну систему;

• виявити впроваджену програмну закладку;

• видалити впроваджену програмну закладку.

При розгляді цих варіантів вирішення завдання захисту від програмних закладок подібно з вирішенням проблеми захисту комп'ютерних систем від вірусів. Як і у випадку боротьби з вірусами, завдання вирішується за допомогою засобів контролю за цілісністю запускаються системних і прикладних програм, а також за цілісністю інформації, що зберігається в комп'ютерній системі і за критичними для функціонування системи подіями. Однак ці кошти дієвими тільки тоді, коли самі вони не схильні до впливу програмних закладок, які можуть:

• нав'язувати кінцеві результати контрольних перевірок;

• впливати на процес зчитування інформації та запуск програм, за якими здійснюється контроль;

• змінювати алгоритми функціонування засобів контролю.

При цьому надзвичайно важливо, щоб включення засобів контролю виконувалося до початку впливу програмної закладки або коли контроль здійснювався тільки з використанням програм управління, що знаходяться в ПЗУ комп'ютерної системи.

Захист від впровадження програмних закладок

Універсальним засобом захисту від впровадження програмних закладок є створення ізольованого комп'ютера. Комп'ютер називається ізольованим, якщо виконані наступні умови:

• в ньому встановлена система BIOS, що не містить програмних закладок;

• операційна система перевірена на наявність в ній закладок;

• достовірно встановлено незмінність BIOS і операційної системи для даного сеансу;

• на комп'ютері не запускалося і не запускається жодних інших програм, окрім вже пройшли перевірку на присутність в них закладок;

• виключений запуск перевірених програм у будь-яких інших умовах. крім перерахованих вище, тобто поза ізольованого комп'ютера.

Для визначення ступеня ізольованості комп'ютера може використовуватися модель ступеневої контролю. Спочатку перевіряється, чи немає змін в BIOS. Потім, якщо все в порядку, зчитується завантажувальний сектор диска і драйверів операційної системи, які, в свою чергу, також аналізуються на предмет внесення до них несанкціонованих змін. І нарешті, за допомогою операційної системи запускається драйвер контролю викликів програм, який стежить за тим, щоб в комп'ютері запускалися тільки перевірені програми.

Цікавий метод боротьби з впровадженням програмних закладок може бути використаний в інформаційній банківській системі, в якій циркулюють виключно файли-документи. Щоб не допустити проникнення програмної закладки через канали зв'язку, в цій системі не допускається прийняття ніякого виконуваного коду. Для розпізнавання подій типу "ОТРИМАНО виконуваний код" і "Отримати файл-ДОКУМЕНТ" застосовується контроль за наявністю у файлі заборонених символів: файл визнається містить виконуваний код, якщо в ньому присутні символи, які ніколи не зустрічаються в файлах-документах.

Виявлення впровадженої програмної закладки

Виявлення впровадженого коду програмної закладки полягає у виявленні ознак його присутності в комп'ютерній системі. Ці ознаки можна розділити на наступні два класи:

• якісні та візуальні;

• виявляються засобами тестування і діагностики.

До якісних і візуальним ознаками відносяться відчуття і спостереження користувача комп'ютерної системи, який відзначає певні відхилення в її роботі (змінюється склад і довжини файлів, старі файли кудись пропадають, а замість них з'являються нові, програми починають працювати повільніше, або закінчують свою роботу дуже швидко, або взагалі перестають запускатися). Незважаючи на те що судження про наявність ознак цього класу здається занадто суб'єктивним, проте, вони часто свідчать про наявність неполадок в комп'ютерній системі і, зокрема, про необхідність проведення додаткових перевірок присутності програмних закладок. Наприклад, користувачі пакету шифрування і цифровий підпис "Кріптоцентр" з деяких пір стали помічати, що цифровий підпис під електронними документами ставиться занадто швидко. Дослідження, проведене фахівцями ФАПСИ, показало присутність програмної закладки, робота якої грунтувалася на нав'язуванні довжини файлу. В іншому випадку тривогу забили користувачі пакета шифрування і цифровий підпис "Криптон", які з подивом відзначили, що швидкість шифрування за криптографічним алгоритмом ГОСТ 28147-89 раптом зросла більш, ніж в 30

разів. А в третьому випадку програмна закладка виявила свою присутність у програмі клавіатурного введення тим, що уражена нею програма перестала нормально працювати.

Ознаки, що виявляються за допомогою засобів тестування та діагностики, характерні як для програмних закладок, так і для комп'ютерних вірусів. Наприклад, завантажувальні закладки успішно виявляються антивірусними програмами, які сигналізують про наявність підозрілого коду в завантажувальному секторі диска. З ініціюванням статичної помилки на дисках добре справляється Disk Doctor, що входить в поширений комплект утиліт Norton Utilities. А засоби перевірки цілісності даних на диску типу Adinf дозволяють успішно виявляти зміни, що вносяться у файли програмними закладками. Крім того, ефективний пошук фрагментів коду програмних закладок за характерними для них послідовностей нулів та одиниць (сигнатурам), а також дозвіл виконанні тільки програм з відомими сигнатурами.

Видалення впровадженої програмної закладки

Конкретний спосіб видалення впровадженої програмної закладки залежить від методу її впровадження в комп'ютерну систему. Якщо це програмно-апаратна закладка, то слід перепрограмувати ПЗУ комп'ютера. Якщо це завантажувальна, драйверной, прикладна, замаскована закладка або закладка-імітатор, то можна замінити їх на відповідну завантажувальну запис, драйвер, утиліту, прикладну або службову програму, отриману від джерела, що заслуговує довіри. Нарешті, якщо це виконуваний програмний модуль, то можна спробувати добути його вихідний текст, прибрати з нього наявні закладки або підозрілі фрагменти, а потім заново відкомпілювати.

2.4.Троянскі програми

Троянською програмою (троянцем, або троянським конем) називається:

• програма яка, будучи частиною іншої програми з відомими користувачеві функціями, здатна потай від нього виконувати деякі додаткові дії з метою заподіяння йому певної шкоди;

• програма з відомими її користувачеві функціями, в яку були внесені зміни, щоб, крім цих функцій, вона могла таємно від нього виконувати деякі інші (руйнівні) дії.

Таким чином, троянська програма - це особливий різновид програмної закладки. Вона додатково наділена функціями, про існування яких користувач навіть не підозрює. Коли троянська програма виконує ці функції, комп'ютерній системі наноситься певних збитків. Однак те, що за одних обставин завдає непоправної шкоди, при інших-може виявитися цілком корисним. Наприклад, програму, яка форматує жорсткий диск, не можна названий. троянської, якщо вона якраз і призначена для його форматування (як це робить команда format операційної системи DOS). Але якщо користувач, виконуючи деяку програму, зовсім не чекає, що вона відформатує його вінчестер, - це і є справжнісінький троянець.

Коротше кажучи, троянською можна вважати будь-яку програму, яка потай від користувача виконує якісь небажані для нього дії. Ці дії можуть бути будь-якими - від визначення реєстраційних номерів програмного забезпечення, встановленого на комп'ютері, до складання списку каталогів на його жорсткому диску. А сама троянська програма може маскуватися під текстовий редактор, під мережеву утиліту або будь-яку програму, яку користувач побажає встановити на свій комп'ютер.

Звідки беруться троянські програми

Троянська програма - це плід праці програміста. Ніяким іншим способом створити її неможливо. Програміст, що пише троянську програму, чудово усвідомлює, чого він хоче добитися, і у своїх намірах він завжди дуже далекий від альтруїзму.

Більшість троянських програм призначено для збору конфіденційної інформації. Їх завдання, частіше за все, полягає у виконанні дій, що дозволяють отримати доступ до даних, які не підлягають широкому розголосу. До таких даних відносяться користувача паролі, реєстраційні номери програм, відомості про банківські рахунки і т. д. Решта троянці створюються для заподіяння прямих збитків комп'ютерній системі, приводячи її у неробочий стан.

До останніх можна віднести, наприклад, троянську програму PC CYBORG, яка заманювала нічого не підозрюють користувачів обіцянками надати їм новітню інформацію про боротьбу з вірусом, що викликає синдром набутого імунодефіциту (СНІД). Проникнувши в комп'ютерну систему, PC CYBORG відраховувала 90 перезавантажень цієї системи, а потім ховала все каталоги на її жорсткому диску і шифрувати знаходяться там файли.

Інша троянська програма називалася AOLGOLD. Вона розсилалася по електронній пошті у вигляді заархівовані файли. У супровідному листі, що додається до цього файлу, йшлося про те, що AOLGOLD призначена для підвищення якості послуг, які надає своїм користувачам найбільший американський Internet-провайдер America Online (AOL). Архів складався з двох файлів, один з яких іменувався INSTALL.BAT. Користувач, що запустив INSTALL.BAT, ризикував стерти всі файли з каталогів C: \, C: \ DOS, C: \ WINDOWS і C: \ WINDOWS \ SYSTEM на своєму жорсткому диску.

Подібного роду троянські програми, як правило, створюються підлітками, які хоч і одержимі пристрастю до руйнування, але не мають глибоких пізнань в програмуванні і тому не можуть заподіяти істотної шкоди комп'ютерним системам, які зазнали нападу створених ними троянців. Наприклад, програма AOLGOLD прала себе з жорсткого диска, будучи запущена з будь-якого іншого дискового розділу за винятком C.

Інша справа - троянські програми, авторами яких є професійні програмісти, які займаються розробкою програмного забезпечення в солідних фірмах. Троянці, що входять до поширені комп'ютерні програми, утиліти та операційні системи, представляють значно більшу загрозу

комп'ютерів, на яких вони встановлені. оскільки їх дії носять не деструктивний характер, а мають на меті збір конфіденційної інформації про систему. Виявити такі троянські програми вдається, як правило, чисто випадково. А оскільки програмне забезпечення, частиною якого вони є, в більшості випадків використовується не тільки якоїсь однієї компанією, закупити це програмне забезпечення, але також на великих Internet-серверах і, крім того, поширюється через Internet, наслідки можуть виявитися жахливими.

Трапляється й так, що троянці вбудовуються в деякі утиліти програмістами, не мають ніякого відношення до розробки цих утиліт. Наприклад, в дистрибутив сканера SATAN, призначений для установки на комп'ютери з операційною системою Linux, що поширювався через Internet, потрапила троянська програма, яка "влаштувалася" в утиліті fping. При першому ж запуску модифікованої утиліти fping в файл / etc / passwd додавалася запис для користувача з ім'ям suser, який в результаті міг увійти в Linux і таємно отримати там повноваження адміністратора. Проте у автора цієї троянської програми були явні прогалини в комп'ютерній освіті. Зокрема, він не знав деяких нюансів зберігання паролів в операційних системах сімейства UNIX. В результаті файл / etc / passwd був відповідним чином змінений лише на двох комп'ютерах, на яких було встановлено цей зіпсований дистрибутив мережевого аналізатора SATAN для Linux.

Де живуть і як часто зустрічаються троянські програми

В даний час троянські програми можна відшукати практично де завгодно. Вони написані для всіх без винятку операційних систем і для будь-яких платформ. Не рахуючи випадків, коли троянські програми пишуться самими розробниками програмного забезпечення, троянці поширюються тим же способом, що і комп'ютерні віруси. Тому найбільш підозрілими на предмет присутності в них троянців, в першу чергу, є безкоштовні і умовно-безкоштовні програми, викачані з Internet, а також програмне забезпечення, яке розповсюджується на піратських компакт-дисках.

Наприклад, в січні 1999 р. було виявлено, що популярна утиліта TCP Wrapper, призначена для адміністрування UNIX-систем і безкоштовно розповсюджується через Internet, на багатьох ftp-сайтах була замінена зовні схожою на неї програмою, яка насправді була троянцем. Після інсталяції він відправляв електронною поштою на певних зовнішніх адресами, оповіщаючи свого господаря про успішне впровадження. Потім він чекав, поки буде встановлено віддалене з'єднання з портом 421 заражених їм комп'ютера, і надавав привілейовані права доступу через цей порт.

Інша троянська програма поширювалася серед користувачів AOL у вигляді вкладення до листа, що розсилаються по електронній пошті. Відкрили це вкладення заражали свій комп'ютер троянцем, який намагався знайти пароль для підключення до AOL і в разі успіху шифрував його. а потім відсилав електронною поштою кудись в Китай.

В даний час існує цілий ряд троянських програм, які можна абсолютно вільно завантажити, підключившись до глобальної комп'ютерної мережі Internet. Найбільшу популярність серед них отримали троянці Back Orifice, Net Bus і SubSeven. На Web-сайті групи розробників Back Orifice, яка іменує себе Cult of Dead Cow (Культ мертвої корови), можна навіть знайти з десяток постерів, які призначені для реклами її останньої розробки - троянця Back Orifice 2000.

Таким чином, троянські програми зустрічаються досить часто і, отже, представляють серйозну загрозу безпеці комп'ютерних систем. Навіть після того як троянська програма виявлена, її шкідливий вплив на комп'ютерну систему може відчуватися ще протягом дуже тривалого часу. Адже найчастіше ніхто не може з упевненістю сказати, наскільки сильно постраждала комп'ютерна система в результаті проникнення в неї троянської програми.

Справа в тому, що більшість троянців є частиною інших програм, які зберігаються в комп'ютері в відкомпілювався вигляді. Текст цих програм являє собою послідовність команд на машинній мові, що складається з нулів і одиниць. Рядовий користувач, як правило, не має ні найменшого поняття про внутрішню структуру таких програм. Він просто запускає їх на виконання шляхом завдання імені відповідної програми в командному рядку або подвійним клацанням на імені її файлу.

Коли з'ясовується, що в якусь скомпільованій проник троянець, в мережі Internet негайно починають поширюватися бюлетені з інформацією про виявлений троянця. Найчастіше в цих бюлетенях коротко повідомляється про те, якої шкоди може завдати даний троянець і де можна знайти заміну ураженої троянцем програмі.

Іноді шкоду, яка може нанести троянець, оцінити досить легко. Наприклад, якщо він призначений для пересилання по електронній пошті вмісту файлу / etc / passwd, в якому операційні системи сімейства UNIX зберігають інформацію про користувача паролі, достатньо встановити "чисту" версію програми замість тієї, в якій влаштувався цей троянець. Потім користувачі повинні будуть відновити свої паролі, і на цьому боротьба з ним успішно завершується.

Однак далеко не завжди ступінь компрометації комп'ютерної системи, в якій оселилася троянська програма, буває так легко визначити. Припустимо, що мета впровадження троянця полягає у створенні діри в захисних механізмах комп'ютерної системи, через яку зловмисник зможе, наприклад, проникати в неї, маючи адміністраторські повноваження. І якщо зломщик виявиться достатньо хитрим і кмітливим, щоб замести сліди своєї присутності в системі шляхом внесення відповідних змін до реєстраційних файли, то визначити, наскільки глибоко він проник крізь системні захисні механізми, буде майже неможливо, якщо врахувати ще той факт, що саму троянську програму виявлять лише кілька місяців після її впровадження в

комп'ютерну систему. У цьому випадку може знадобитися цілком перевстановити операційну систему і всі програми.

Як розпізнати троянську програму

більшість програмних засобів, призначених для захисту від троянських програм, в тій чи іншій мірі використовує так зване узгодження об'єктів. При цьому в якості об'єктів фігурують файли і каталоги, а узгодження є спосіб відповісти на питання. чи змінилися файли і каталоги з моменту останньої перевірки. У ході узгодження характеристики об'єктів порівнюються з характеристиками. якими вони володіли раніше. Береться, наприклад, архівна копія системного файлу і її атрибути порівнюються з атрибутами цього файлу, який зараз знаходиться на жорсткому диску. Якщо атрибути розрізняються і ніяких змін в операційну систему не вносилося, значить в комп'ютер, швидше за все, проник троянець.

Одним з атрибутів будь-якого файлу є відмітка про час його останньої модифікації: щоразу, коли файл відкривається, змінюється і зберігається на диску, автоматично вносяться відповідні поправки. Однак позначка часу не може служити надійним індикатором наявності в системі троянця. Справа в тому, що нею дуже легко маніпулювати. Можна підкрутити тому системний годинник, внести зміни в файл, потім знову повернути годинник в початковий стан, і відмітка про час модифікації файлу залишиться незмінною.

Може бути, інакше йде справа з розміром файлу? Аж ніяк. Нерідко текстовий файл, який спочатку займав, скажімо, 8 Кбайт дискового простору, після редагування та збереження має той самий розмір. Трохи інакше поводяться виконавчі файли. Вкласти в чужу програму фрагмент власного коду так, щоб вона не втратила працездатності і в відкомпілювався вигляді зберегла свій розмір, досить непросто. Тому розмір файлу є більш надійним показником, ніж відмітка про час внесення до нього останніх змін.

Зловмисник, який вирішив запустити в комп'ютер троянця, зазвичай намагається зробити його частиною системного файлу. Такі файли входять в дистрибутив операційної системи та їх присутність на будь-якому комп'ютері, де ця операційна система встановлена, не викликає жодних підозр. Проте будь-який системний файл має цілком певну довжину. Якщо цей атрибут буде якимось чином змінений, це стривожить користувача.

Знаючи це, зловмисник постарається дістати вихідний текст відповідної програми і уважно проаналізує його на предмет присутності в ньому надлишкових елементів, які можуть бути видалені без жодного відчутного збитку. Тоді замість знайдених надлишкових елементів він вставить у програму свого троянця і перекомпілюється її заново. Якщо розмір отриманого двійкового файлу виявиться менше або більше розміру початкового, процедура повторюється. І так до тих пір, поки не буде отриманий файл, розмір якого найбільшою мірою близький до оригіналу (якщо вихідний файл досить великий, цей процес може розтягнутися на кілька днів).

Отже, у боротьбі з троянцями покластися на відмітку про час останньої модифікації файлу і його розмір не можна, оскільки зловмисник може їх досить легко підробити. Більш надійною в цьому відношенні є так звана контрольна сума файлу. Для її підрахунку елементи файлу підсумовуються, і вийшло в результаті число оголошується його контрольною сумою. Наприклад, в операційній системі SunOS існує спеціальна утиліта sum, яка виводить на пристрій стандартного висновку STDOUT контрольну суму файлів, перелічених у рядку аргументів цієї утиліти.

Однак і контрольну суму в загальному випадку виявляється не так вже й важко підробити. Тому для перевірки цілісності файлової системи комп'ютера використовується особливий різновид алгоритму обчислення контрольної суми, звана одностороннім хешем.

Функція хешування називається односторонньою, якщо завдання відшукання двох аргументів, для яких її значення збігаються, є труднорешаемой. Звідси випливає, що функція одностороннього хешування може бути застосована для того, щоб відстежувати зміни, внесені зловмисником у файлову систему комп'ютера, оскільки спроба зловмисника змінити будь-який файл так, щоб значення, отримане шляхом одностороннього хешування цього файлу, залишилося незмінним, приречена на невдачу .

Історично склалося так, що більшість утиліт, що дозволяють боротися з проникненням в комп'ютерну систему троянських програм шляхом односпрямованого хешування файлів, було створено для операційних систем сімейства UNIX. Однією з найбільш зручних в експлуатації і ефективних є утиліта Trip Wire, яку можна знайти в Internet за адресою http://www.tripwiresecurity.com/. Вона дозволяє робити односпрямоване хешування файлів за допомогою декількох алгоритмів, в тому числі - MD43, MD54 і SHA5. Обчислені хеш-значення файлів зберігаються в спеціальній базі даних, яка, в принципі, є найбільш уразливим ланкою \ утиліти TripWire. Тому користувачам TripWire пропонується в обов'язковому порядку приймати додаткові заходи захисту, щоб виключити доступ до цієї бази даних з боку зловмисника (наприклад, поміщати її на знімному носії, призначеному тільки для читання).

Засоби боротьби з троянцями в операційних системах сімейства Windows (95/98/NT) традиційно є частиною їх антивірусного програмного забезпечення. Тому, щоб відловлювати Back Orifice, Net Bus, SubSeven та інші подібні до них троянські програми, необхідно обзавестися найсучаснішим антивірусом (наприклад, програмою Norton Anuvirus 2000 компанії Symantec , яка дозволяє виявляти присутність в комп'ютерній системі найбільш поширених троянців і позбуватися від них). Слід регулярно перевіряти свій комп'ютер на присутність в ньому вірусів.

Тим, хто хоче мати в своєму розпорядженні утиліту, призначену саме для виявлення троянців в комп'ютерах, що працюють під управлінням

операційних систем сімейства Windows, можна порадити звернути свої погляди на програму The Cleaner компанії MooSoft Development (http://www.homestead.com/ moosoft / cleaner.html) Ця утиліта може бути з успіхом використана для боротьби з більш ніж чотирма десятками різновидів троянських програм.

Огляд засобів боротьби з троянськими програмами був би далеко не повним, якщо обійти увагою нещодавно з'явилися на ринку програмні пакети, призначені для комплексної захисту від загроз, з якими стикаються користувачі настільних комп'ютерів при роботі в Internet. Одним з таких пакетів є eSafe Protect компанії Aladdin Knowledge Systems (демонстраційну версію eSafe Protect можна знайти в Internet no адресою www.esafe.com).

Функціонально eSafe Protect ділиться на три компоненти - антивірус, персональний брандмауер і модуль захисту комп'ютерних ресурсів. Антивірус позбавляє комп'ютер від шкідливих програм завдяки застосуванню антивірусного модуля VisuSafe, сертифікованого американським Національним агентством комп'ютерної безпеки. Персональний брандмауер контролює весь вхідний і вихідний трафік по протоколу TCP / IP, наділяючи використовуються IP-адреси певними правами (наприклад, обмежуючи доступ в Internet в певні години або забороняючи відвідини деяких Web-сайтів).

Для захисту ресурсів комп'ютера, на якому встановлений програмний пакет eSafe Protect, створюється спеціальна ізольована область - так звана пісочниця. Всі автоматично завантажуються з Internet Java-аплсти і компоненти ActiveX спочатку поміщаються в "пісочницю", де вони знаходяться під невсипущим наглядом eSafe Protect. Якщо потрапила в "пісочницю" програма спробує виконати будь-яке недозволене дію, то воно буде негайно блоковано. Протягом заданого інтервалу часу (від 1 до 30 днів) кожен додаток, завантажене в комп'ютер з Internet. проходить "карантинну" перевірку в "пісочниці". Отримана в холі такої перевірки інформація заноситься в особливий журнал. Після закінчення "карантину" додаток буде виконуватися поза "пісочниці", проте йому будуть дозволені тільки ті дії, перелік яких визначається на основі наявних журнальних записів.

Таким чином, у порівнянні з іншими подібними програмними пакетами eSafe Protect забезпечує найбільш розвинені і ефективні засоби комплексної зашиті від троянських програм. Вхідний до складу eSafe Protect антивірус допомагає швидко виявляти троянців і розправлятися з ними, використовуючи технології, які добре зарекомендували себе при боротьбі з вірусами. Персональний брандмауер блокує будь-які спроби зв'язатися ззовні з проникли в комп'ютерну систему троянськими програмами. І нарешті з допомогою "пісочниці" своєчасно запобігає впровадження троянців в комп'ютери під виглядом Java-аплетів і компонентів ActiveX.

2.5.Клавіатурні шпигуни

Одна з найбільш поширених різновидів програмних закладок - клавіатурні шпигуни. Такі програмні закладки націлені на перехоплення

паролів користувачів операційної системи, а також на визначення їх легальних повноважень і прав доступу до комп'ютерних ресурсів.

Клавіатурні шпигуни - явище зовсім не нове в світі комп'ютерів. Свого часу вони розроблялися і для OS/370, і для UNIX, і для DOS. Їх поведінка в загальному випадку є досить традиційним: типовий клавіатурний шпигун обманним шляхом заволодіває користувацькими паролями, а потім переписує ці паролі туди, звідки їх може без особливих зусиль витягти зловмисник. Відмінності між клавіатурними шпигунами стосуються тільки способу, який застосовується ними для перехоплення призначених для користувача паролів. Відповідно всі клавіатурні шпигуни поділяються на три типи - імітатори, фільтри та заступники.

Імітатори

Клавіатурні шпигуни цього типу працюють за наступним алгоритмом. Зловмисник впроваджує в операційну систему програмний модуль, що імітує запрошення користувачеві зареєструватися для того, щоб увійти в систему. Потім впроваджений модуль (у прийнятій термінології - імітатор) переходить в режим очікування введення користувацького ідентифікатора і пароля. Після того як користувач ідентифікує себе і введе свій пароль, імітатор зберігає ці дані там, де вони доступні зловмисникові. Далі імітатор ініціює вихід із системи (що в більшості випадків можна зробити програмним шляхом), і в результаті перед очима у нічого не підозрює користувача з'являється сто одне, але на цей раз вже справжнє запрошення для входу в систему.

Обдурений користувач, бачачи, що йому пропонується ще раз внести пароль. приходить до висновку про те, що він допустив якусь помилку під час попереднього введення пароля, і слухняно повторює всю процедуру входь в систему заново. Деякі імітатори для переконливості видають на екран монітора правдоподібне повідомлення про нібито досконалої користувачем помилково. Наприклад, таке: "НЕВІРНИЙ ПАРОЛЬ. СПРОБУЙТЕ ЩЕ РАЗ".

Написання імітатора не вимагає від його творця будь-яких особливих навичок. Зловмиснику, який вміє програмувати на одному з універсальних мов програмування (наприклад, на мові BASIC), знадобляться на це лічені години. Єдина складність, з якою він може зіткнутися, полягає в тому, щоб відшукати в документації відповідну програмну функцію, що реалізує вихід користувача з системи.

Перехоплення пароля часто полегшують самі розробники операційних систем, які не ускладнюють себе створенням ускладнених за формою запрошень користувачеві зареєструватися для входу в систему. Подібне зневажливе ставлення характерно для більшості версій операційної системи UNIX, в яких реєстраційне запрошення складається з двох текстових рядків, які видаються по черзі на екран термінала:

login:

password:

Щоб підробити таке запрошення, не потрібно бути семи п'ядей у лобі. Однак саме по собі ускладнення зовнішнього вигляду запрошення не створює для хакера, який задумав впровадити в операційну систему імітатор, яких-небудь непереборних перешкод. Для цього потрібно вдатися до більш складних і витонченим заходам захисту. Як приклад операційної системи, в якій такі заходи у досить повному обсязі реалізовані на практиці, можна привести Windows NT.

Системний процес WinLogon, що відповідає в операційній системі Windows NT за аутентифікацію користувачів, має свій власний робочий стіл - сукупність вікон, одночасно видимих на екрані дисплея. Цей робочий стіл називається столом аутентифікації. Ніякий інший Процес, в тому числі і імітатор, не має доступу до робочого столу аутентифікації і не може розташувати на ньому своє вікно.

Після запуску Windows NT на екрані комп'ютера виникає так зване початкове вікно робочого столу аутентифікації, що містить вказівку натиснути на клавіатурі клавіші <Ctrl> + <Alt> + <Del>. Повідомлення про натискання цих клавіш передається тільки системному процесу WinLogon, а для інших процесів, зокрема, для всіх прикладних програм, їх натискання відбувається абсолютно непомітно. Далі виробляється перемикання на інше, так зване реєстраційне вікно робочого столу аутентифікації. У ньому-то якраз і розміщується запрошення користувачеві ввести своє ідентифікаційне ім'я та пароль, які будуть сприйняті і перевірені процесом WinLogon.

Для перехоплення користувацького пароля впроваджений в Windows NT імітатор обов'язково повинен вміти обробляти натискання користувачем клавіш <Ctrl> + <Alt> + <Del>. В іншому випадку відбудеться переключення на реєстраційне вікно робочого столу аутентифікації, імітатор стане неактивним і не зможе нічого перехопити, оскільки всі символи пароля, введені користувачем, минуть імітатор і стануть надбанням виключно системного процесу WinLogon. Як вже говорилося, процедура реєстрації в Windows NT влаштована таким чином, що натискання клавіш <Ctrl> + <Alt> + <Del> проходить безслідно для всіх процесів, крім WinLogon, і тому для користувача пароль надійде саме йому.

Звичайно, імітатор може спробувати відтворити не початкове вікно робочого столу аутентифікації (в якому висвічується вказівку користувачеві одночасно натиснути клавіші <Ctrl> + <Alt> + <Del>), а реєстраційне (де міститься запрошення ввести ідентифікаційне ім'я та пароль користувача). Однак за відсутності імітаторів в системі реєстраційне вікно автоматично замінюється на початкове після короткого проміжку часу (залежно від версії Window NT він може тривати від 30 с до 1 хв), якщо протягом цього проміжку користувач не робить ніяких спроб зареєструватися в системі. Таким чином, сам факт занадто довгої присутності на екрані реєстраційного вікна повинен насторожити користувача Windows NT і змусити його ретельно перевірити свою комп'ютерну систему на предмет наявності в ній програмних закладок.

Підводячи підсумок сказаному, можна відзначити, що ступінь захищеності Windows NT від імітаторів досить висока. Розгляд захисних механізмів, реалізованих в цій операційній системі, дозволяє сформулювати дві необхідні умови, дотримання яких є обов'язковим для забезпечення надійного захисту від імітаторів:

• системний процес, який при вході користувача в систему отримує від нього відповідні реєстраційне ім'я і пароль, повинен мати свій власний робочий стіл, недоступний іншим процесам;

• переключення на реєстраційне вікно робочого столу аутентифікації має відбуватися абсолютно непомітно для прикладних програм, які до того ж ніяк не можуть вплинути на це переключення (наприклад, заборонити його).

На жаль, ці дві умови ні в одній з операційних систем, за винятком Windows NT, не дотримуються. Тому для підвищення їх захищеності від імітаторів можна порекомендувати скористатися адміністративними заходами. Наприклад, зобов'язати кожного користувача негайно повідомляти системного адміністратора, коли вхід в систему виявляється неможливий з першого разу, незважаючи на коректно задане ідентифікаційне ім'я та правильно набраний пароль. Фільтри

Фільтри "полюють" за всіма даними, які користувач операційної системи вводить з клавіатури комп'ютера. Самі елементарні фільтри просто скидають перехоплений клавіатурний ввід на жорсткий диск або в якесь інше місце, до якого має доступ зловмисник. Більш, витончені програмні закладки цього типу піддають перехоплені дані аналізу і фільтрують інформацію, що має відношення до призначених для користувача паролів.

Фільтри є резидентними програмами, перехоплювачем одне або декілька переривань, які пов'язані з обробкою сигналів від клавіатури. Ці переривання повертають інформацію про самій клавіші і введеному символі, що аналізується фільтрами на предмет виявлення даних, що мають відношення до пароля користувача.

Відомі кілька фільтрів, створених спеціально для різних версій операційної системи DOS. У 1997 р. відзначена поява фільтрів для операційних систем Windows 3.11 і Windows 95.

Треба сказати, що виготовити подібного роду програмну закладку не становить великої праці. В операційних системах Windows 3.11 і Windows 95/98 передбачений спеціальний програмний механізм, за допомогою Якого в них вирішується ряд завдань, пов'язаних з отриманням доступу до введення з клавіатури, в тому числі і проблема підтримки національних розкладок клавіатур. Приміром, будь-клавіатурний русифікатор для Windows являє собою самий що ні на є справжній фільтр, оскільки покликаний перехоплювати всі дані, що вводяться користувачем з клавіатури комп'ютера. Неважко "допрацювати" його таким чином, щоб разом зі своєю основною функцією (підтримка національної розкладки клавіатури) він заодно виконував би і дії з перехоплення паролів. Тим більше, що але. багатьох навчальних посібниках і довідниках користувача операційних систем

Windows є вихідні тексти програмних русифікаторів клавіатури. "Перепрофілювали" цей русифікатор так, щоб він узяв на себе виконання функцій клавіатурного шпигуна, його можна вбудувати перед цим русифікатором або після нього, і в результаті вся інформація. 4 вводиться користувачем з клавіатури, піде і через клавіатурного шпигуна. Таким чином завдання створення фільтра стає такою простою, що не вимагає наявності будь-яких спеціальних знань у зловмисника. Йому залишається тільки непомітно впровадити виготовлену їм програмну закладку в операційну систему і вміло замаскувати її присутність.

У загальному випадку можна стверджувати, що якщо в операційній системі дозволяється перемикати клавіатурну розкладку під час введення пароля, то для цієї операційної системи можливе створення фільтра. Тому, щоб убезпечити її від фільтрів, необхідно забезпечити виконання наступних трьох умов:

• під час введення пароля переключення розкладок клавіатури не дозволяється;

• конфігурувати ланцюжок програмних модулів, що беруть участь в роботі з паролем користувача, може лише системний адміністратор;

• доступ до файлів цих модулів має виключно системний адміністратор.

Дотримати перше з цих умов у локалізованих версіях операційних систем принципово неможливо. Справа в тому, що засоби створення облікових записів користувача російською мовою є невід'ємною частиною таких систем. Тільки в англомовних версіях систем Windows NT і UNIX передбачені можливості, що дозволяють підтримувати рівень безпеки, при якій дотримуються всі 3 перераховані умови.

Заступники

Заступники повністю або частково підміняють собою програмні модулі операційної системи, що відповідають за аутентифікацію користувачів. Подібного роду клавіатурні шпигуни можуть бути створені для роботи в середовищі практично будь багатокористувацької операційної системи. Трудомісткість написання заступника визначається складністю алгоритмів, реалізованих підсистемою аутентифікації, і інтерфейсів між її окремими модулями. Також при оцінці трудомісткості слід брати до уваги ступінь документованості цієї підсистеми. В цілому можна сказати, що завдання створення заступника значно складніше завдання написання імітатора або фільтра. Тому фактів використання подібного роду програмних закладок зловмисниками поки відмічено не було. Однак у зв'язку з тим, що в даний час все більшого поширення набуває операційна система Windows NT, що має потужні засоби захисту від імітаторів і фільтрів, в самому недалекому майбутньому від хакерів слід очікувати більш активного використання заступників з метою отримання несанкціонованого доступу до комп'ютерних систем.

Оскільки заступники беруть на себе виконання функцій підсистеми аутентифікації, перед тим як приступити до перехоплення призначених для користувача паролів вони повинні виконати наступні дії:

• подібно комп'ютерному вірусу впровадитися в один або декілька системних файлів;

• використовувати інтерфейсні зв'язку між програмними модулями полсістеми аутентифікації для вбудовування себе в ланцюжок обробки введеного користувачем пароля.

Для того щоб захистити систему від впровадження заступника, її адміністратори повинні строго дотримуватися адекватну політику безпеки. І що особливо важливо, підсистема аутентифікації повинна бути одним з найбільш захищених елементів операційної системи. Однак, як показуємо практика, адміністратори, подібно всім людям, схильні до скоєння помилок. А отже, дотримання адекватної політики безпеки протягом необмеженого періоду часу є нездійсненним завданням. Крім того, як тільки заступник потрапив у комп'ютерну систему, будь-які заходи захисту від впровадження програмних закладок перестають бути адекватними, і тому необхідно передбачити можливість використання ефективних засобів виявлення та вилучення впроваджених клавіатурних шпигунів. Це означає, що адміністратор повинен вести дуже ретельний контроль цілісності виконуваних системних файлів і інтерфейсних функцій, використовуваних підсистемою аутентифікації для вирішення своїх завдань.

Але і ці заходи можуть виявитися недостатньо ефективними. Адже машинний код заступника виконується в контексті операційної системи, і тому заступник може вживати особливих заходів, щоб максимально утруднити власне виявлення. Наприклад, він може перехоплювати системні виклики, що використовуються адміністратором для виявлення програмних закладок, з метою підміни повертається ними інформації. Або фільтрувати повідомлення, що реєструються підсистемою аудиту, щоб відсіювати ті, які свідчать про його присутність у комп'ютері.

Як захистити систему від клавіатурних шпигунів

Клавіатурні шпигуни представляють реальну загрозу безпеці сучасних комп'ютерних систем. Щоб відвести цю загрозу, потрібно реалізувати цілий комплекс адміністративних заходів та програмно-апаратних засобів захисту. Надійний захист від клавіатурних шпигунів може бути побудована тільки тоді, коли операційна система має певними можливостями, що утрудняють роботу клавіатурних шпигунів. Вони були докладно описані вище, і не має сенсу знову на них зупинятися.

Однак необхідно ще раз відзначити, що єдиною операційною системою, в якій побудова такого захисту можливо, є Windows NT.

Та й то із застереженнями, оскільки все одно її доведеться забезпечити додатковими програмними засобами, що підвищують ступінь її захищеності. Зокрема, в Windows NT необхідно ввести контроль цілісності системних файлів і інтерфейсних зв'язків підсистеми аутентифікації.

Крім того, для надійного захисту від клавіатурних шпигунів адміністратор операційної системи повинен дотримуватися політику безпеки, при якій тільки адміністратор може:

• конфігурувати ланцюжка програмних модулів, що беруть участь в процесі аутентифікації користувачів;

• здійснювати доступ до файлів цих програмних модулів;

• конфігурувати саму підсистему аутентифікації.

І нарешті, при організації захисту від клавіатурних шпигунів завжди слід мати на увазі, що ні неухильне дотримання адекватної політики безпеки, ні використання операційної системи, що має в своєму складі засоби, істотно ускладнюють впровадження клавіатурних шпигунів і полегшують їх своєчасне виявлення, ні додаткова реалізація контролю за цілісністю системних файлової та інтерфейсних зв'язків самі по собі не можуть служити запорукою надійного захисту інформації в комп'ютері. Всі ці заходи повинні здійснюватися в комплексі. Адже жертвою клавіатурного шпигуна може стати будь-який користувач операційної системи, оскільки її адміністратори теж люди, час від часу і вони допускають помилки в своїй роботі, а для впровадження клавіатурного шпигуна достатньо всього однієї помилки адміністратора.

3. Парольний захист операційних систем

Контроль доступу, заснований на володінні специфічною інформацією, найбільш поширений. Він характеризується тим, що правом доступу мають лише ті, хто здатний продемонструвати знання певного секрету, звичайно пароля. Це найпростіший і дешевий метод захисту будь-якої комп'ютерної системи. Оскільки його використання не вимагає великих витрат часу, сил і місця в пам'яті комп'ютера, то він застосовується навіть у тих комп'ютерах, які зовсім не потребують засобах захисту. Крім того, володіння паролем дає користувачеві відчуття психологічного комфорту. Більш того, це широко використовується в системах, вже захищених іншими засобами - магнітними картками або іншими програмними засобами, типу шифрування, що в ще більшій мірі підвищує рівень захисту від несанкціонованого доступу. До цього часу єдиним засобом захисту комп'ютерної мережі від несанкціонованого доступу була парольна система. При стандартній процедурі входу в мережу кожен користувач повинен знати своє ім'я мережі і мережевий пароль. Адміністратор, що призначає ці атрибути, як правило, не застосовує випадкових чи погано запам'ятовуються послідовностей символів, оскільки це може призвести до того, що мережеве ім'я та пароль можуть бути записані на будь-які носії (папір, дискету і т. п.), що може привести до витоку секретного пароля та імені користувача. Паролі, як правило, розглядаються в якості ключів для входу в систему, але

вони використовуються і для інших цілей: блокування запису на дисковод, в командах на шифрування даних, тобто у всіх тих випадках, коли потрібно тверда впевненість, що так діяти будуть тільки законні власники або користувачі програмного забезпечення. І донині в багатьох випадках для зловмисника основним (іноді єдиним) захисним кордоном проти атак в комп'ютерній мережі залишається система парольного захисту, яка є у всіх сучасних операційних системах. Відповідно до встановленої практики перед початком сеансу роботи з операційною системою користувач зобов'язаний зареєструватися, повідомивши їй своє ім'я та пароль. Ім'я потрібно операційній системі для ідентифікації користувача, а пароль служить підтвердженням правильності виробленої ідентифікації. Інформація, введена користувачем в діалоговому режимі, порівнюється з тією, яка є в розпорядженні операційної системи. Якщо перевірка дає позитивний результат, то користувачеві будуть доступні всі ресурси операційної системи, повязані з його ім'ям. Важко уявити, що сьогодні якомусь зловмисникові може прийти в голову шалена думка про те, щоб спробувати підібрати ім'я та пароль для входу в операційну систему, по черзі перебираючи в умі, всі можливі варіанти і вводячи їх з клавіатури. Швидкість такого підбору пароля буде надзвичайно низькою, тим більше, що в операційних системах з добре продуманою парольного захистом кількість поспіль повторних введень конкретного користувача імені і відповідного йому пароля завжди можна обмежити двома-трьома і зробити так, що якщо це число буде перевищено, то вхід в систему з використанням даного імені блокується протягом фіксованого періоду часу або до приходу системного адміністратора. Тому частіше використовують більш небезпечний і набагато більш ефективний метод злому парольного захисту операційної системи, при використанні якого атаці піддається системний файл, що містить інформацію про її легальних користувачів і їх паролі.

Однак будь-яка сучасна операційна система надійно захищає користувача паролі, які зберігаються в цьому файлі за допомогою шифрування. Доступ до таких файлів за замовчуванням заборонений, як правило, навіть для системних адміністраторів, не кажучи вже про рядових користувачів. Іноді зловмисникові вдається шляхом різних хитрувань отримати в своє розпорядження файл з іменами користувачів і їх зашифрованими паролями. І тоді йому на допомогу приходять спеціалізовані програми - парольні зломщики, які і служать для злому паролів операційних систем. Як же діють ці програми? Криптографічні алгоритми, застосовувані для шифрування паролів користувачів в сучасних операційних системах, в переважній більшості випадків занадто стійкі для того, щоб можна було сподіватися відшукати методи їх дешифрування, які виявляться більш ефективними, ніж тривіальний перебір можливих варіантів. Тому парольні зломщики іноді просто шифрують всі паролі з використанням того ж самого криптографічного алгоритму, який застосовується для їх засекречування в

атакується операційній системі, і порівнюють результати шифрування з тим, що записано в системному файлі, де знаходяться шифровані паролі її користувачів. При цьому в якості варіантів паролів парольні зломщики використовують символьні послідовності, автоматично генеруються з деякого набору символів. Даним способом можна зламати всі паролі, якщо відомо їх подання в зашифрованому вигляді і вони містять тільки символи з цього набору. Максимальний час, необхідний для злому пароля, залежить від кількості символів в наборі, граничної довжини пароля і від продуктивності комп'ютера, на якому проводиться злом її парольного захисту (залежить від операційної системи і швидкодії). Зі збільшенням числа символів у вихідному наборі, число перебираються комбінацій зростає експоненціально, тому такі атаки парольного захисту операційної системи можуть займати занадто багато часу. Однак добре відомо, що більшість користувачів операційних систем не ускладнюють себе вибором стійких паролів (тобто таких, які важко зламати). Тому для більш ефективного підбору паролів парольні зломщики зазвичай використовують так звані словники, що представляють собою заздалегідь сформований список слів, найбільш часто вживаних як паролів. Для кожного слова зі словника парольний зломщик використовує одне або кілька правил. Відповідно до цих правил слово змінюється і породжує додаткове безліч опробуемих паролів. Проводиться почергове зміна літерного регістра, в якому набрано слово, порядок проходження букв у слові змінюється на зворотний, на початок і в кінець кожного слова приписується цифра 1, деякі літери замінюються на близькі по зображенню цифри (в результаті, наприклад, зі слова password виходить pa55wOrd). Це підвищує ймовірність підбору пароля, оскільки в сучасних операційних системах, як правило, розрізняються паролі, набрані великими та малими літерами, а користувачам цих систем настійно рекомендується вибирати паролі, в яких букви чергуються з цифрами. Протистояти таким атакам можна лише втом випадку, якщо використовувати стійкі до злому паролі. Перед тим як відповісти на запитання «Як правильно вибрати пароль», розглянемо, які ж паролі використовуються взагалі. Паролі можна поділити на сім основних груп: паролі, що встановлюються користувачем; паролі, що генеруються системою; випадкові коди доступу, що генеруються системою; півслова; ключові фрази; інтерактивні послідовності типу «питання-відповідь»; «Суворі» паролі. Перша група найбільш поширена. Більшість таких паролів відносяться до типу «вибери сам». Для кращого захисту від несанкціонованого доступу необхідно використовувати досить довгий пароль, тому зазвичай система запитує пароль, що містить не менше чотирьох-п'яти букв. Існують також і інші заходи, що не дозволяють користувачеві створити невдалий пароль. Наприклад, система може наполягати на тому, щоб пароль включав в себе малі та великі літери упереміш з цифрами; завідомо очевидні паролі, наприклад, internet, нею відкидаються. У різних операційних системах існує чимало програм, які переглядають файли, що містять паролі, аналізують

паролі користувачів і визначають, наскільки вони секретні. Невідповідні паролі замінюються.

Коли людина вперше завантажує комп'ютер, і той запитує у нього пароль, цей пароль напевно виявиться варіантом однієї із загальних і актуальних для всіх тем - особливо якщо у користувача не вистачає часу. Не рахуючи геніїв і безнадійних тупиць, всі люди, коли треба приймати швидкі рішення, мислять і діють приблизно однаково. І користувачі видають перше, що приходить їм у голову. А в голову приходить те, що вони бачать чи чують у даний момент, або те, що збираються зробити відразу ж після завантаження. У результаті пароль створюється в поспіху, а подальша його заміна на більш надійний відбувається досить рідко. Таким чином, багато паролі, створені користувачами, можна розкрити досить швидко. Випадкові паролі і коди, що встановлюються системою, бувають кількох різновидів. Системне програмне забезпечення може використовувати повністю випадкову послідовність символів, аж до випадкового вибору регістрів, цифр, пунктуації довжини; або ж використовувати обмеження у генеруючих процедурах. Створювані комп'ютером паролі можуть також випадковим чином вилучатись зі списку звичайних або нічого не значущих слів, створених авторами програми, які утворюють паролі на кшталт onah.foopn, або ocar-back-treen. Півслова частково створюються користувачем, а частково - будь-яким випадковим процесом. Це означає, що якщо навіть користувач придумає легко вгадується пароль, наприклад, «абзац», комп'ютер доповнить його якою-небудь плутаниною, утворивши більш складний пароль типу «абзац, 3ю37». Ключові фрази хороші тим, що вони довгі і їх важко вгадати, зате легко запам'ятати. Фрази можуть бути осмисленими, типу «ми були стурбовані цим» або не мати сенсу, наприклад, «ловить рибу ніс». Слід зауважити, що в програмуванні поступово намічається тенденція до переходу на більш широке застосування ключових фраз. До концепції ключових фраз близька концепція кодового акроніма, який експерти з захисту оцінюють як коротку, але ідеально безпечну форму пароля. У акроніми користувач бере легко запам'ятовується пропозицію, фразу, рядок з вірша і т. п., і використовує перші літери кожного слова в якості пароля. Наприклад, акроніми двох наведених вище фраз є «мбое» і «ЛРН». Подібні нововведення в теорії паролів значно ускладнюють заняття електронним шпигунством. Інтерактивні послідовності «питання-відповідь», пропонують користувачеві відповісти на кілька питань, як правило, особистого плану: «Дівоче прізвище вашої матері?», «Ваш улюблений колір?", І т. д. У комп'ютері зберігаються відповіді на безліч таких питань. При вході користувача в систему комп'ютер порівнює отримані відповіді з «правильними». Системи з використанням «питання-відповідь» схильні переривати роботу користувача кожні десять хвилин, пропонуючи відповідати на питання, щоб підтвердити його право користуватися системою. В даний час такі паролі майже не застосовуються. Коли їх

придумали, ідея здавалася непоганий, але дратівливий чинник переривання привів до того, що даний метод практично зник з побуту. «Суворі» паролі зазвичай використовуються спільно з яким-небудь зовнішнім електронним або механічним пристроєм. У цьому випадку комп'ютер зазвичай з простодушним підступністю пропонує кілька варіантів запрошень, а користувач повинен дати на них відповідні відповіді. Паролі цього типу часто зустрічаються в системах з одноразовими кодами. Одноразові коди - це паролі, які спрацьовують тільки один раз. До них іноді вдаються, створюючи тимчасову копію для гостей, щоб продемонструвати потенційним клієнтам можливості системи. Вони також часом застосовуються при першому входженні користувача в систему. Під час першого сеансу користувач вводить свій власний пароль і надалі входить у систему лише через нього. Одноразові коди можуть також застосовуватися в системі, коли дійсний користувач входить в неї в перший раз, потім вам слід поміняти свій пароль на більш секретний персональний код. У випадках, коли системою користується група людей, але при цьому не можна порушувати таємність, вдаються до списку одноразових кодів. Той чи інший користувач вводить код, відповідний 'час, дату або дню тижня. Отже, для того щоб пароль був дійсно надійний, він повинен відповідати певним вимогам: бути певної довжини; включати в себе великі та малі літери; включати в себе одну і більше цифр; включати в себе один нецифровий і один неалфавітний символ. Одне або декілька з цих правил повинні обов'язково дотримуватися. Необхідно пам'ятати, що пароль - це сама слабка частина будь-якої системи захисту даних, якою б гострою та надійної вона не була. Саме тому його вибору і зберігання треба приділити належну увагу. Не варто спокушатися і тішитися своєю безпекою при роботі з Windows 95/98, якщо бачите, в будь-якому діалоговому вікні ваш пароль, прихований зірочками - це захист «від дурня». За допомогою крихітної програми можна подивитися прихований зірочками пароль, всього лише встановивши курсор миші на діалогове вікно

4. Безпека даних у комп’ютерних мережах

Однією з головних проблем, що виникає під час проектування, встановлення та експлуатації комп’ютерної мережі, є безпека даних, оскільки перевагою мережі є доступ до спільних даних та пристроїв, а це зумовлює можливість несанкціонованого доступу до них.

Безпека даних – це захист ресурсів мережі від руйнування та захист даних від випадкового чи навмисного розголошення, а також від неправочинних змін. Рівні безпеки даних.

1.Законодавство

2.Адміністративний контроль

3.Фізичні засоби захисту

4.Вбудовані засоби

5.База даних

У сучасних системах захист даних реалізується на багатьох рівнях :

• вбудовані засоби захисту – програмно-системні (паролі, права доступу та ін.);

• фізичні засоби захисту – замки, двері, охорона, сигналізація тощо;

• адміністративний контроль – організаційні заходи, накази адміністрації;

• законодавство та соціальне оточення – соціальний клімат колективу, нетерпимість до несанкціонованого використання чужої інформації, комп'ютерного "піратства", закони про захист авторських та майнових прав. У кожній інформаційній системі можна виділити найслабші з погляду безпеки місця. На них адміністратор повинен звернути увагу передусім. До таких місць, як звичайно, належать: сховища даних, адміністративна система, кабельна система, доступ з зовнішніх мереж. Долають труднощі, пов'язані з безпекою даних, одночасно у трьох напрямках:

• профілактика; мінімізація ймовірності настання небажаних подій; унеможливлення несанкціонованого доступу; профілактика апаратури;

• якщо небажана подія сталася, система повинна бути побудована так, щоб мінімізувати шкоду, якої ця подія завдасть;

• створення процедур архівації та поновлення інформації у випадку її втрати.

Як же на практиці відбувається надання та обмеження прав доступу? Найпростіше описати цей механізм використанням таблиць чинності. Таблиця чинності ставить у відповідність певній категорії об'єктів операційної системи набір прав доступу до ресурсів мережі: створення, використання, управління ресурсом тощо. Об'єктами можуть бути:

• окремі користувачі чи групи користувачів;

• ступінь таємності;

• прикладні програми;

• час доби;

• робоча станція;

• довільна комбінація цих об'єктів (контейнер).

Такий підхід дає змогу гнучко формулювати складні обмеження доступу (наприклад, дозволити доступ до каталогу з розважальними програмами тільки на час обідньої перерви або з певних робочих станцій). Чинні права доступу для користувача, які формуються як комбінація обмежень з таблиць чинності, називаються ефективними правами доступу цього користувача. У деяких системах (наприклад, банківських чи податкових) потрібна ідентифікація не користувача, а фізичної особи. Розрізняють кілька способів такої ідентифікації:

• за персональними фізичними ознаками (біометрія). Знімають відбиток пальця, а потім ідентифікують чинність особи. Інший спосіб: система пропонує вголос повторити певну кількість випадково вибраних слів

та аналізує особливості голосу. Такі системи досить надійні, однак значно дорожчі за традиційні;

• за предметом, який особа – користувач носить з собою. Таким предметом може бути спеціальний значок, магнітна картка з кодом. Цей спосіб є дешевим, проте ненадійним, предмет можна підробити, вкрасти тощо;

• за тим, що особа повинна знати або пам’ятати. Треба пам’ятати пароль або правильно відповісти на низку питань. Цей метод найдешевший і найпоширеніший, але ненадійний (пароль можна підібрати, відповіді вгадати). Інформаційний захист мережі від зовнішніх втручань (intrusions) здійснюється з використанням брандмауерів та серверів–посередників (proxy–серверів).

4.1. Технології з’єднань комп’ютерів

Фізичне підключення двох ПК

Більшість ПК має один або декілька послідовних портів. Ці порти можна використовувати для підключення будь–якого пристрою з інтерфейсом RS–232–C і для зв'язку або управління. В даному розділі ми розглянемо, як підключити інтерфейс RS–232–C для забезпечення зв'язку типу ПК – ПК, термінал – ПК і модем – ПК. Почнемо з розгляду базової моделі RS–232–C, показаної на рис. 2. Ця модель ілюструє, як можуть з’єднуватися один з одним два ПК і/або термінали через модеми по телефонних лініях або прямим зв'язком. Хоча подальше обговорення ми ведемо переважно в термінах телефонних з'єднань, ті ж базові принципи відносяться і до прямого зв'язку, за винятком того, що комунікаційні пристрої (DCE, Data Communication Equipment) в цьому випадку не потрібні.

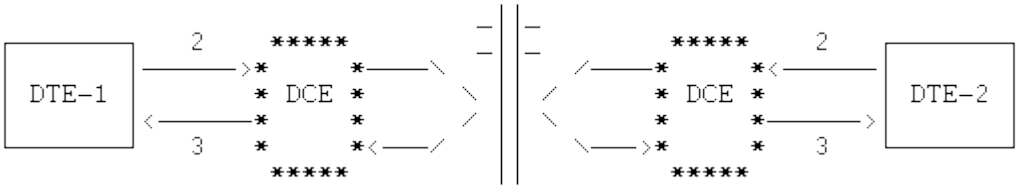

Рис. 2. Стандартна модель інтерфейсу RS–232–C.

На кожному кінці знаходяться термінальні пристрої, так звані DTE (Data Terminating Equipment). В ролі DTE може виступати термінал, наприклад, VT–100, або центральний процесор мікро, міні або великої ЕОМ. Кожний термінальний пристрій DTE повинен використовувати комунікаційний пристрій DCE (Data Communication Equipment), який зазвичай називають модемом, для **мо**дуляції і **дем**одуляції сигналів, що проходять по телефонних лініях. Кожний DTE використовує вивід 2 для передачі даних і вивід 3 для отримання даних. Оскільки те, що передано з виводу 2 на кожній машині, приймається на виводі 3 іншої машини, виникає перехрещення телефонних ліній між пристроями DCE. Під'єднання і обробка сигналу між DTE і DCE повністю відповідають стандарту RS–232–C. Апаратний протокол дозволяє DTE використовувати DCE для посилки і

прийому даних від іншого DTE. Кабель, що зв'язує фізично DTE і DCE, називається "прямим" кабелем. Він дозволяє пристрою DTE посилати команди (або сигнали з виводів) на DCE, а пристрою DCE відправляти команди назад на DTE. Підключення DCE однієї машини до DCE іншої машини проводиться через звичайні телефонні лінії. Пристрої DCE необхідні з тієї причини, що пристрої DTE є цифровими, а телефонні лінії – аналоговими. Єдиний спосіб передати цифрову інформацію по аналогових лініях – закодувати цифрову інформацію в аналоговий сигнал, послати цей сигнал по телефонних лініях, а потім декодувати аналоговий сигнал назад в цифрову інформацію.

Підключення без комунікаційних пристроїв

Якщо ваші машини розташовані досить близько (в межах 15 метрів), вам не потрібен модем, ви можете використовувати кабель "нуль–модему" замість DCE. Кабель нуль–модему імітує такий же протокол, що і DCE, але не вимагає наявності модему для комунікацій. Основна задача підключення нуль–модему забезпечити перехрещення між передаючими і приймаючими сигналами. На рис. 3 показана загальна схема підключення без пристроїв DCE.

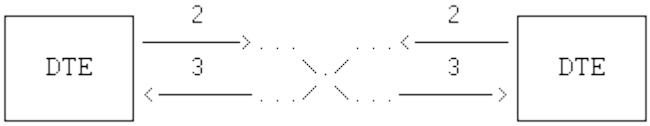

Рис. 3. Конфігурація з’єднання нуль–модемом

Для того, щоб виконати підключення, яке імітує DCE, потрібні деякі маніпуляції з сигналами. Ці маніпуляції також стандартизовані в кабелі нуль–модему. По схемі цього кабелю, показаній на рис. 4, розглянемо, як він імітує сигнали DCE. Лінії 1 і 7 використовуються для шасі і сигнальної землі відповідно. Лінії 2 і 3 перетинаються так, щоб коли одна сторона говорить, інша слухала. Обидві сторони можуть говорити одночасно (режим Full Duplex), якщо використовувати різні набори дротів. Для імітації управляючих сигналів лінії 4, 5 і 8 приєднуються так, щоб кожного разу, коли пристрій DTE–1 активізує лінію "Request To Send" ("запит на передачу"), тобто передає по ній сигнал, він одержує назад сигнал "Clear To Send" ("готовий до передачі"), який вказує на те, що інша сторона готова прийняти дані. Потім, посилаючи сигнал по лінії "Data Carrier Detect" ("виявлення потоку даних"), пристрій DTE–1 повідомляє іншу сторону, що поступають дані. Таке методичне "апаратне рукостискання" гарантує, що ніякі дані не будуть відправлені, поки

інша сторона не буде готова їх прийняти.

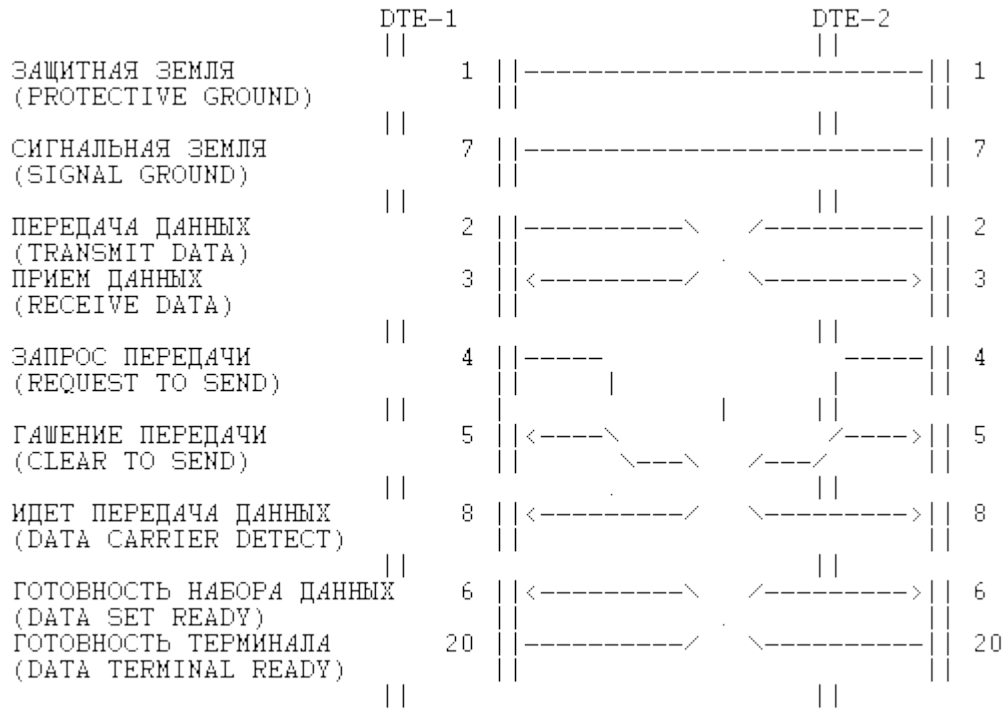

Рис. 4. Кабель нуль–модему RS–232–C.

Лінії 6 і 20 приєднуються так, щоб забезпечити решту
1 7 2 3 4 5 8 6 20 1 7 2 3 4 5 8 6 20
Protective Ground
(захисне заземлення)
Signal Ground
(сигнальне заземлення)
Transmit Data
(передача даних)
Receive Data
(прийом даних)
Request To Send
(запит на передачу)
Clear To Send
(готовий до передачі)
Data Carrier Detect
(виявлення потоку даних)
Data Set Ready
(готовність даних)
Data Terminal Ready
(готовність терміналу)

управляючих сигналів нуль–модему. Поки DTE активний ("Data Terminal Ready" – "готовність терміналу", лінія 20), інша сторона вважає, що має справу з активним модемом ("Data Set Ready" – "готовність даних", лінія 6). При такому способі з'єднання ліній 6 і 20 кожного разу, коли висмикнути кабель з ПК або перемкнути його на інший канал сполучної коробки, інша сторона втрачає сигнал активності і відключається (або генерує сигнал HUP –

Hangs UP, повісити трубку). Щоб зробити такий кабель, який не викликає відключення при вийманні штепселя (тобто NOHUP), слід приєднати вихід "Data Terminal Ready" до входу "Data Set Ready" на тому ж пристрої DTE. Це примушує систему повідомляти самій собі, що модем завжди готовий. Зауважимо, що хоча розглянута схема підключення нуль–модему є рекомендованою, але існують й інші способи. У кожному конкретному випадку для нуль–модемів враховується певне оточення або функція, наприклад, наявність безобривного (nohup) варіанту підключення. Розглянемо способи комунікацій і типи підключення, які найчастіше використовуються.

Дистанційне підключення

Альтернативою прямому підключенню є дистанційне підключення через модемну лінію. Установка терміналу або конфігурація ПЕОМ виглядають приблизно так само, як і у попередньому випадку, за винятком швидкості обміну, на якій працює термінал. Для більшості модемів вона повинна бути не менша 1200 бод. Термінал (коли він встановлений на 1200 бод) спілкується безпосередньо з модемом. При цьому задіяні модемні команди "набрати телефонний номер" (dial), "повісити трубку" (hang) і т.д. ПЕОМ, що запускає комунікаційне програмне забезпечення, звичайно має команду набору номера, яка генерує команду для модему. З'єднання між терміналом/ПЕОМ і модемом повинно бути виконано у вигляді прямого кабелю. Модем має також телефоний кабель, що йде в телефонну систему.

Підключення по виділеній лінії

Одна з технологій для одночасної голосової телефонії, інтерактивного відео і передачі даних з великою швидкістю – FTTN (Fiber all the way To The Home) – це високо – швидкісна цифрова абонентська лінія, або VDSL (Very high rate Digital Subscriber Line). VDSL передає дані з великою швидкістю по витій парі мідних телефонних ліній, не приєднаних до апаратури АТС (виділена лінія), з рядом швидкостей, залежних від фактичної довжини лінії. Максимальна швидкість у 55 Мб/с досягається для ліній довжиною до 300 метрів та 13 Мб/с – до 1500 метрів .

VDSL знаходиться все ще в стадії вивчення; не досить відомі особливості впливу телефонної лінії, радіоемісія, придатність для мультиплексного з’єднання та інформаційні вимоги для створення набору стандартних властивостей.

Локальні мережі множинного доступу

Безумовно, найпопулярнішим стандартом на мережі з магістральною організацією передавального середовища (множинного доступу) з тих, що використовуються на сьогодні є стандарт Ethernet (IEEE 802.3) . Мережі Ethernet працюють з швидкістю 10 Мбіт/с або 100 Мбіт/с (новий стандарт – Fast Ethernet). Ethernet пропонує краще співвідношення продуктивність вартість, високу гнучкість при налаштуванні і нарощуванні потужностей, а також відносну простоту в експлуатації. Ethernet використовує метод передачі даних, відомий як CSMA/CD (Carrier Sense Multiple Access with

Collision Detection – метод доступу з прослуховуванням несучої і виявленням зіткнень). Перед відправкою даних мережею вузол спочатку прослуховує, чи не посилає в даний момент інформацію який–небудь інший вузол. Якщо ж по мережі передається якась інформація, вузол, який намагається відправити дані, чекає якийсь час і потім знову повторює спробу. Існує декілька типів Ethernet, що використовуються сьогодні:

- 10Base5 (Thicknet – товстий Ethernet)
- 10Base2 (Thinnet – тонкий Ethernet)
- 10BaseT (UTP – вита пара)
- 100BaseTX, 100BaseT4 (Fast Ethernet)
- 100BaseFX (оптоволокно).

Дозволяється змішувати різні стандарти Ethernet. Наприклад, якщо концентратори, які використовуються як точки центрального з'єднання для комп'ютерів в мережі UTP Ethernet також містять порт BNC, це дозволяє використовувати сегмент кабелю тонкого Ethernet (thinnet) для з'єднання різних концентраторів. Основні правила, встановлені між двома вузлами мережі Ethernet:

- може бути сполучено до п'яти мережних сегментів підряд;
- може бути до чотирьох повторювачів / концентраторів;
- може бути до трьох заповнених сегментів (в даному випадку тонкий Ethernet).

Наявні в лабораторії типи мереж Ethernet: **10Base2** (Thinnet), або тонкий Ethernet, – це дуже відомий тип Ethernet (особливо, для малих мереж). Тонкий Ethernet використовує топологію шини, що складається з коаксіального кабелю RG58A/U з навантаженням 50 Ом на кожному кінці (термінатором). Комп'ютери підключаються до сегменту кабелю тонкого Ethernet за допомогою Т – подібних BNC – конекторів, що вставляються безпосередньо в плату мережного адаптера Ethernet. До тонкого Ethernet застосовуються наступні правила:

- максимальна довжина кабельного сегменту (відстань між двома обмежувачами – термінаторами з навантаженням по 50 Ом) не більше 185 м;
- кожний сегмент мережного кабелю повинен мати навантаження в 50 Ом на обох кінцях;
- максимальне число вузлів на сегмент не більше 30;
- довжина відрізка кабелю між мережними адаптерами не менше 2м;
- максимальне число вузлів в мережі не більше 1024;
- максимальна відстань між двома вузлами не більше 1425 м.

10BaseT/UTP (Unshielded Twisted Pair – неекранована вита пара), або просто вита пара, призначена для нових мереж. UTP використовує топологію зірки, де кожний вузол підключається до концентратора (Hub), або розподільника (Switch). Розподільник є центральною точкою з'єднання; можна об'єднати декілька розподільників. Кабель, що використовується для UTP, складається з двох неекранованих витих пар і часто називається кабелем категорії 3; він

підтримує швидкість до 10 Мбіт/с (кабель категорії 5 також може використовуватися і він з відповідним устаткуванням може підтримувати швидкість до 100 Мбіт/с). До витої пари застосовуються наступні правила:

• довжина кабельного сегменту між вузлом і розподільником не більше 100 м;

• в з'єднанні RJ–45 використовуються напряму сполучені контакти 1, 2, 3 і 6, причому контакти 1 і 2 – передаючі, а контакти 3 і 6 – приймаючі;

• до центрального розподільника може бути підключено до 12 інших розподільників;

• в мережі витої пари може бути не більше 1024 робочих станцій (без використання мостів).

100BaseTX, або 100BaseT чи швидкий Ethernet (Fast Ethernet), за топологією подібний 10BaseT, з тією лише різницею, що він працює із швидкістю 100 Мбіт/с замість 10 Мбіт/с. 100BaseT, як і 10BaseT, використовує дві неекрановані виті пари, але для підтримки швидкості в 100 Мбіт/с вимагає кабелю категорії 5 і строгого дотримання стандартів обтискання. Для роботи зі швидкістю 100 Мбіт/с мережний адаптер, як і концентратор, повинен підтримувати 100BaseTX. 100BaseTX також підтримує більш повільний стандарт (10 Мбіт/с), в результаті можна використовувати 100BaseTX – адаптери в мережі 10BaseT (100BaseTX – адаптери працюватимуть із швидкістю 10 Мбіт/сзамість 100 Мбіт/с).

4.2. Інформаційний захист мережі з використанням брандмауерів та серверів–посередників Інформаційний захист мережі від зовнішніх втручань здійснюється з використанням брандмауерів та серверів–посередників .

Первинне значення терміна брандмауер (firewall) – це стіна у будівлі, зроблена з вогнетривких та незаймистих матеріалів, яка повинна перешкодити поширенню пожежі. У комп'ютерній мережі **брандмауер** – це комп'ютер з програмною системою, який ставлять на межі внутрішньої мережі і який перепускає тільки авторизовані Tower box Tower box

Інтернет

Локальна

мережа

Сервер-посередник Брандмауер

певним чином пакети.

Найчастіше брандмауери захищають внутрішню корпоративну мережу від несанкціонованого доступу із зовнішньої мережі. Однак їх можна використовувати для фільтрування вихідної інформації, обмеження доступу користувачів внутрішньої мережі назовні. Брандмауери застосовують різні алгоритми фільтрування, вони мають різні ступені захисту та вартість. З метою класифікації брандмауерів їхню роботу описують з використанням семи рівнів еталонної моделі взаємодії відкритих систем (OSI) .

Розрізняють:

• брандмауери з фільтруванням пакетів (Packet Filtering Firewall), які працюють на канальному та мережному рівнях);

• шлюзи рівня сеансу (Circuit Level Gateway); працюють на рівні розпізнання сеансу;

• шлюзи рівня застосувань (Application Level Gateway); фільтрують інформацію згідно програмних застосувань);

• брандмауери експертного рівня (Stateful Inspection Firewall); виконують функції брандмауерів усіх нижніх рівнів.Як правило, чим вищий рівень роботи брандмауера, тим більший рівень захисту, який він забезпечує і тим більша його вартість. Брандмауери з фільтруванням пакетів працюють разом з апаратним або програмним маршрутизатором. Вони аналізують IP – заголовки пакетів і на підставі інформації у них та своєї таблиці правил приймають рішення про проходження пакета чи його відкидання. Брандмауери з фільтруванням пакетів порівняно дешеві та вносять невелику затримку у передавання повідомлень. Часто функції фільтрування пакетів інтегрують у маршрутизатори. Водночас рівень захисту у таких брандмауерів незначний – кваліфікований зловмисник може підмінити адресну частину IP

пакета. В ідеальному випадку брандмауер повинен бути прозорим (непомітним) для клієнтів мережі. Це означає, що він не спричинює суттєвої затримки в передаванні інформації, не вимагає від клієнтів спеціальної реєстрації на брандмауері, відокремленої від реєстрації користувача в мережній ОС. На практиці вимога прозорості брандмауера тою чи іншою мірою порушується. Інколи функції брандмауера в складних системах розподілені між власне брандмауерами та серверами – посередниками (proxy серверами). У чому ж різниця між цими серверами? Брандмауер традиційно захищає мережу від зовнішніх втручань. Він фільтрує кадри канального рівня, розпізнає сеанс, який відкриває зовнішній користувач. Сервер – посередник контролює та обмежує вихід внутрішнього користувача назовні, а також часто є його представником. Функції сервера – посередника такі:

• приховати адреси внутрішніх станцій, подаючи всю мережу назовні як один комп’ютер з адресою сервера;

• зберігати популярні web–сторінки, файли, так що користувачі не змушені звертатися до зовнішньої мережі при їх повторному запиті. Популярну інформацію сервер оновлює автоматично з визначеною періодичністю.

Конфігурація брандмауера в ОС UNIX

У UNIX легко встановити заснований на правилах фільтрування IP–пакетів мережний захист [11]. Можна захистити тільки один ПК або цілу мережу.

Типи мережного захисту повинні бути "клієнтом" (“client”), щоб забезпечити єдину автономну машину, або "простий" (“simple”) для входу, що охороняє внутрішню мережу.

Важливе зауваження: мережний захист UNIX розробляєтьсямаксимально безпечним за замовчуванням. Отож, якщо ви не

додаєте ніяких правил, то заборонені будуть **всі** пакети. Може виявитись, що неможливо дістатися до вашої машини через мережу і доведеться реєструватися з консолі ПК. Мережний захист також запобігає новим зв'язкам із зовнішньою частиною мережі (за винятком декількох протоколів, як наприклад, електронна пошта), що унеможливлює такі звичні мережні протоколи як FTP, telnet тощо. Може виявитись, що вам не сподобається консервативний набір правил за замовчуванням. Якщо це так, легко зробити ваш власний. Перша річ, яку ви можете зробити, це – дозволити зв'язки через **ssh** (**ssh** – це безпечна заміна **telnet** / **rlogin**, її можна знайти за адресою http://www.openssh.org/). Там, де в наборі правил говориться "Дозволити поступати електронній пошті", слід додати подібне правило для **ssh** за допомогою заміни номера порта 25 на 22. Або можна зробити повністю новий набір правил. Тут наведено два фрагменти наборів правил для типового клієнта ("client") і для мережі ("simple"):

Набір правил firewall для окремого ПК:

Встановити IP адресу сервера

ip="194.44.198.193"

setup_loopback

Дозволити весь вихідний потік із сервера

$fwcmd add allow all from $ip to any out

Заборонити вихідний потік з будь–яких інших адрес

$fwcmd add deny log all from any to any out

Дозволити пакети для яких вже встановлено TCP з'єднання

$fwcmd add allow tcp from any to any established

Дозволити фрагментовані IP пакети

$fwcmd add allow all from any to any frag

Дозволити пакети, які ініціюють з'єднання ftp, ssh, email, tcp–dns, http

$fwcmd add allow tcp from any to $ip 21 setup

$fwcmd add allow tcp from any to $ip 22 setup

$fwcmd add allow tcp from any to $ip 25 setup

$fwcmd add allow tcp from any to $ip 53 setup

$fwcmd add allow tcp from any to $ip 80 setup

Набір правил Firewall для мережі:

Опис зовнішнього інтерфейсу

oif="fxp0"

onet="194.44.198.192"

omask="255.255.255.224"

oip="194.44.178.193"

Опис внутрішнього інтерфейсу

iif="fxp1"

inet="192.168.2.19"

imask="255.255.255.0"

iip="192.168.2.119"

setup_loopback

Трансляція мережних адрес (natd) розміщена після правил перевірки # адрес з тим, щоб пакети зі станцій із внутрішньої мережі (192.168.x.x) # транслювались natd після того, як вони будуть відкинуті правилами # заборони (deny) перед цим.

case $natd_enable in
[Yy][Ee][Ss])
if [–n "$natd_interface"]; then
$fwcmd add divert natd all from any to any via
$natd_interface
fi ;;
esac
Дозволити всі пакети у внутрішній мережі
$fwcmd add allow all from any to any via $iif
Дозволити всі пакети назовні
$fwcmd add allow all from $onet:$omask to any out via $oif
Заборонити всі пакети назовні з інших підмереж
$fwcmd add deny log all from any to any out via $oif
Дозволити пакети для яких вже встановлено TCP з’єднання
$fwcmd add allow tcp from any to any established
Дозволити фрагментовані IP пакети
$fwcmd add allow all from any to any frag
Все решту заборонити і протоколювати
*$fwcmd add deny log all from any to any_*

4.3. Захист ресурсів в мережній ОС Novel NetWare Мережа NetWare – це група ПК (файл–серверів та робочих станцій) і принтерів, які з’єднані разом так, що їх користувачі можуть використовувати спільні ресурси. Файл–сервер – це ПК, на якому працює операційна система NetWare, яка керує мережею. Файл–сервер координує роботу всіх робочих станцій і регулює, хто з користувачів може мати доступ до потрібних ресурсів, хто може змінювати дані. Можливість працювати в мережі отримують замовники, попередньо зареєстровані як користувачі мережі. Є чотири рівні доступу до ресурсів мережі:

• звичайні користувачі мережі;
• оператори файл–серверів;
• менеджери підгруп та менеджери обліку;
• адміністратори мережі.

Вся інформація мережі NetWare зберігається на жорсткому диску, який знаходиться на файл–сервері. Тим не менше, не всі користувачі можуть мати доступ до повної інформації (наприклад, до файлів обліку). Також користувачі не завжди можуть одночасно мати доступ до одних і тих же даних, бо інакше вони впливатимуть на роботу один одного. Щоб запобігти таким проблемам, NetWare передбачає потужну систему безпеки, яка захищає користувачів від пошкодження даних в мережі і унеможливлює

несанкціонований доступ до заборонених файлів. Система безпеки NetWare складається з таких компонент :

• система реєстрації (керує обліковими записами, які включають імена користувачів, їх паролі та набір прав і обмежень користувача в мережі);

• систему привілеїв, що надають користувачам дозвіл працювати з ресурсами;

• атрибути, призначені каталогам і файлам. Система Netware забезпечує високий ступінь захисту інформації, збереженої на мережних томах. Система захисту NetWare здійснює контроль за тим,

• хто може звертатися до мережних каталогів;

• до яких каталогів і файлів можуть звертатися користувачі;

• що користувачі можуть робити з каталогами і файлами;

• хто може виконувати задачі на консолі файл–сервера.

Захист в ОС Netware має три рівні:

• захист входу користувача в систему;

• захист за допомогою схеми прав власності;

• захист за допомогою схеми атрибутів.

Захист входу в систему керує доступом до ресурсів мережі: тут визначається, які користувачі можуть працювати на файл сервері, коли вони можуть працювати, на яких робочих станціях і які ресурси можуть використовувати. Мережний адміністратор встановлює захист входу в систему, використовуючи три інструментальні засоби:

• usernames (імена користувачів);

• passwords (паролі);

• restrictions (обмеження).

Тільки мережні адміністратори і менеджери робочих групп можуть створювати нові імена користувачів. Всім новоствореним користувачам призначені паролі і членство в групі EVERYONE . Використовуються три типи обмежень входу в систему:

• з яких робочих станцій може входити в систему даний користувач;

• в який робочий час користувачі можуть увійти в систему;

• якщо перевищені квоти робочого простору чи ін., обліковий запис користувача блокується системою аж до втручання адміністратора мережі.

Захист за допомогою схеми прав власності визначає до яких каталогів, підкаталогів і файлів користувач має доступ і як саме він може ними розпоряджатися. Захист правами визначається довірчими правами та маскою

успадкованих прав, які містять однакові вісім атрибутів:

• контролю (Supervisory, S)

• читання (Read, R)

• запису (Write, W)

• створення (Create, C)

• вилучення (Erase, E)

• зміни (Modify, M)

• перегляд файлу (File scan, F)
• контроль доступу (Access control, A)

Сервісні програми Netware відображають початкові символи цих прав у дужках: [S R W C E M F A]. Щоб переглянути ефективні права для каталогу чи файлу, використовуються команди RIGHTS або WHOAMI. Захист за допомогою схеми атрибутів полягає у призначенні спеціальних властивості індивідуальним каталогам або файлам, які перекривають довірчі права і можуть запобігати діям, які б дозволяли ефективні права. Атрибути можуть запобігати від вилучення, копіювання, модифікації чи перегляду файлу або каталогу. Атрибути також використовуються для контролю за спільним використанням ресурсів (shared), маркуванням модифікованих файлів для того, щоб утиліти резервного копіювання могли вибирати тільки змінені файли, а також для запобігання перекрученням файлів (corruption). ОС Netware використовує наступні атрибути каталогів:

• заборона вилучення (Delete Inhibit, D)
• прихований (Hidden, H)
• очищення (Purge, P)
• заборона перейменування (Rename Inhibit, R)
• системний (System, Sy),

а також наступні атрибути файлів:

• підлягає архіву (Archive Needed, A)
• заборона копіювання (Copy Inhibit, C)
• заборона видалення (Delete Inhibit, D)
• тільки для виконання (Execute Only, X)
• прихований (Hidden, H)
• Індексований (Indexed, I)
• очищення (Purge, P)
• аудит читання (Read Audit, Ra)
• тільки для читання (Read Only, Ro)
• читання / запис (Read Write, Rw)
• заборона перейменування (Rename Inhibit, R)
• спільний (Shareable, S)
• системний (System, Sy)
• переміщуваний (Transactional, T)
• аудит запису (Write Audit, Wa).

Зауваження:

А. Якщо користувачі мають право модифікації каталогу чи файлу, вони можуть змінювати атрибути і виконувати будь–яку задачу, дозволену їх ефективними правами.

В. Атрибути каталогів і файлів слід використовувати для посилення захисту там, де багато користувачів мають доступ до файлів. Приклад: утиліти ОС Netware так захищено атрибутами, що навіть СУПЕРВІЗОР не може видаляти їх без того, щоб зняти спочатку відповідні прапорці.

C. Усі файли ОС Netware у системних каталогах SYS:SYSTEM, SYS:PUBLIC і SYS:LOGIN мають атрибути Ro, S, D, і R. D. Файли бази даних користувачів мають атрибути Sy, H, і T. E. Для зміни чи перегляду атрибутів використовують утиліти FILER, FLAG, or FLAGDIR

Система простежування транзакцій (TTS)

Система простежування транзакцій (переміщень даних) ОС Netware (Transaction Tracking System, TTS) захищає прикладні програми від перекручування, виконуючи зворотнє трасування (backing out) незавершених транзакцій, які виникають при відмові мережних компонент. При зворотньому трасуванні, дані й індексна інформація в базі даних повертаються до того стану, у якому вони були, перш, ніж почалася транзакція. TTS – невід'ємна частина ОС Netware v3.x; навіть при відсутності бази даних для багатьох користувачів на вашому сервері; вона реалізована на рівні операційної системи файл сервера. Перевагою такого підходу є те, що навіть прикладні програми, спеціально не розроблені для оперативного повернення отримують такі можливості. TTS захищає дані при невдачі, роблячи копію первісних даних перш, ніж записати поверх нові дані. Якщо невдача відбувається протягом транзакції, TTS відновлює первісні дані. TTS може захищати проти цих типів невдач будь–які прикладні програми, що допускають запити із блокуванням записів і зберігають інформацію в записах на жорсткому диску. Файли обробки текстів, що не організовані в дискретні записи, не захищені TTS. Перелічимо наступні типи потенційних проблем захисту:

A. Користувачі, що були зроблені еквівалентом супервізора.

B. Користувачі, що мають небезпечні паролі або не мають ніяких.

C. Користувачі, що мають довірчі права в кореневому каталозі будь–якого тому.

D. Користувачі, що мають права на SYS:SYSTEM.

4.4. Захист електронної пошти

Електронна пошта чи пошта e–mail – один з найпопулярніших видів використання Інтернету. За допомогою електронної пошти в Інтернеті ви можете відіслати лист мільйонам людей по всій планеті і одночасно листи отримати. Електронна пошта стає усе більш важливою умовою ведення повсякденної та ділової діяльності. Основні групи загроз, що походять від електронного листування, це:

- троянські коні;
- віруси;
- програми зловмисного характеру, що містяться в прикріплених до листів файлах;
- поштові віруси – черв’яки (Melissa, Back Door, Sobig та ін.);
- спамові листи.

Для безпечного листування потрібно встановити антивірусну програму, яка має у своєму складі резидентний модуль, який постійно зберігається в пам'яті комп'ютера і яка відловлює всі підозрілі рухи поштового клієнта. Вибір тут достатньо великий, але рекомендують Антивірус Касперського (AVP) або Данилова (DrWeb) з регулярним поновленням антивірусних баз. Програми надійні і забезпечують захист від усіх видів комп'ютерних вірусів і

троянських коней. Можуть бути небезпечними електронні листи, які не містять ніяких вкладень, зате уражені так званими скрипт – вірусами (поштовими вірусами – черв'яками). Серед найвідоміших, слід, зокрема, згадати KakWorm, Stages і ILOVEYOU (LoveLetter). Вони написані на Visual Basic for Applications (VBA), використовують Windows Scripting Host (машину для запуску скрипт – програм) і вкрай небезпечні. При цьому, проти них часто безсилі традиційні антивірусні засоби, які не в змозі знайти присутність віруса, якщо він не звертається до жорсткого диску, а оперує виключно в оперативній пам'яті комп'ютера. Евристичний аналізатор AVP Script Checker, спеціально призначений для боротьби зі скрипт – вірусами. Перед виконанням скриптів Checker проводить евристичний аналіз коду і його перевірку за допомогою AVP Монітора. При виявленні віруса або підозрілого коду на екран буде виведено відповідне попередження і скрипт не буде виконаний. Вільно поширювану програму MailCleaner, також призначену для боротьби зі скрипт – вірусами:, можна отримати за адресою: *ftp://www.mailcleaner.com/MCSetup.exe*

Дуже оперативно програми, призначені для боротьби з конкретними видами вірусів, з'являються на сайті

http://www.computerra.ru/scallwin/www.download.com, в розділі Downloads : PC : Utilities : Antivirus. Для боротьби зі спамовими листами розроблено протокол безпечної електронної пошти (SSL). Більшість клієнтів електронної

пошти можна сконфігурувати для безпечних операцій з електронною поштою з використанням SSL. На вкладці Advanced в MS Outlook Express, наприклад, є опція: "This server requires a secure connection (SSL)". Поштовий клієнт автоматично встановить нові порти для такого з'єднання. Іноді слід задати ці порти вручну. Дозволені такі порти SSL для наявних поштових послуг: SMTP– SSL (465), POP3–SSL (995) та IMAP–SSL (993).

Сервер електронної пошти використовує непідписане посвідчення SSL для безпечного кодування операцій пересилання електронної пошти. Зазвичай видається попередження про "непідписане посвідчення" при пересилці і отриманні електронної пошти, але процес залишається безпечним і шифрованим. Те ж справедливо і для клієнта webmail, який доступний з HTTPS. Цифрове посвідчення SSL – це електронний файл, який унікально ідентифікує індивідуумів і сервери. Цифрові посвідчення дозволяють клієнту (мережний навігатор) засвідчити сервер до установки сеансу SSL. Звичайно, цифрові посвідчення підписуються незалежною і довіреною третьою

стороною, щоб гарантувати їх переконливість. Сторона, що підписала документ, називається уповноваженою (CA) стороною, як наприклад VeriSign. При повсюдному використанні протоколу SSL стане можливим заборонити передавання та прийом непідписаної кореспонденції, що дозволить уникнути небажаного спамового засилля. А тим часом в боротьбі зі спамом допомагають ряд спеціально призначених для цього програм:

• Telos версії 2.0 сканує вміст поштових ящиків на предмет виявлення заголовків спамових листів (які листи віднести до спаму, визначає сам користувач) і видаляє їх. Програма може працювати в автоматичному режимі, скануючи поштові ящики через певні проміжки часу. Telos можна налаштувати і для знищення листів, що приходять з певних адрес:*http://members.xoom.com/bsoft/telos200.zip*

• SpammerSlammer, який працює фільтром між вашою поштовою програмою і POP3 –сервером, відсікає небажані листи (критерії небажаності визначаються самим користувачем): *http://sb.nowtools.com/spammerslammer.exe*

• Якщо троянському коневі все–таки вдалося якимось чином пробратися на ваш комп'ютер, то вашу систему допоможе захистити програма Jammer чи Ad–Aware. Вони відстежують спроби запису в реєстр і встановлення з'єднання між серверною і клієнтською частинами трояна: *http://www.agnitum.com/download/jammer.exe*

5.Основи криптографії

Під *криптографією* будемо розуміти область знань, що відноситься до методів і засобів перетворення повідомлень у незрозумілу для сторонніх осіб форму, а також перевірки істинності цих повідомлень. Під *криптоаналітикою* будемо розуміти засоби і методи, спрямовані на подолання криптографічного захисту. Сукупність криптографії та криптоаналітики називається *криптологією*. *Розшифровуванням* будемо називати відновлення вихідного повідомлення при відомому ключі шифрування. *Дешифруванням* будемо називати процес відновлення вихідного повідомлення при невідомому ключі шифрування. Таким чином, ті, кому призначено шифроване повідомлення його *розшифровують*, а ті, хто перехоплює його, намагаються *дешифрувати*.

Клод Шеннон у своїй роботі „Теорія связи в секретных системах" узагальнив накопичений до нього досвід розробки шифрів. Вияснилося, що навіть у дуже складних шифрувальних системах можна виділити в якості складових частин шифри заміни, шифри перестановки та їх комбінації. Деякі відомості про ці прості шифри можна знайти у художній літературі, зокрема „Золотий жук" Едгара По і „Пляшущие человечки" Артура Конан Дойла.

Розглянемо два приклади простих шифрів.

Шифр „Сцитала". Цей шифр відомо з часів війни Спарти проти Афін у V ст. до н.е. Для його реалізації використовувалась т.зв. „сцитала" – циліндричний жезл певного діаметру. На сциталу намотували вузьку

папірусну стрічку і на ній писали повідомлення вздовж осі сцитали. Коли стрічку знімали, на ній залишалися незрозумілі літери. Для розшифровки повідомлення адресат намотував стрічку на такий самий жезл і читав повідомлення. В цьому шифрі перетворення оригінального тексту у шифрований зводиться до перестановки літер оригінального тексту. Тому клас таких шифрів отримав назву *шифру перестановки*.

Шифр Цезаря. Цей шифр реалізує таке перетворення відкритого тексту: кожна літера замінюється третьою після неї літерою алфавіту, який вважається написаним по колу, тобто після „я" йде „а". Відмітимо, що Цезарь замінював її третьою літерою, але можна міняти і будь-якою іншою. Головне, щоби адресат цього повідомлення знав величину і напрямок цього зсуву. Клас шифрів, до якого відноситься шифр Цезаря, називається *шифрами заміни*. З цього, напевне зрозуміло, що створення хорошого шифру є задачею непростою. Тому бажано збільшити час життя шифру, але тут зростає імовірність того, що криптоаналітики противника зможуть розкрити шифр і читати зашифровані повідомлення. Якщо у шифрі є змінний „ключ", то його заміна призводить до того, що розроблені противником методи вже не дадуть ефекту. Під *ключем* будемо розуміти змінний елемент шифру, який застосовується для шифрування конкретного повідомлення. Наприклад, у шифрі сцитала ключем є діаметр жезлу, а у шифрі Цезаря – величина і напрямок зсуву літер шифротексту відносно літер відкритого. Описані міркування призвели до того, що безпека повідомлень, що шифруються, в першу чергу стала забезпечуватися ключем. Сам шифр, шифромашина або принцип шифрування прийнято вважати відомими суперникові і доступними для попереднього вивчення, але у шифрі з'явився невідомий елемент –ключ, від якого істотно залежать застосовувані перетворення інформації. Тепер користувачі, перш ніж обмінятися шифрованими повідомленнями, повинні обмінятися ключем, за допомогою якого можна прочитати зашифроване повідомлення. А для криптоаналітиків, які хочуть прочитати перехоплене повідомлення, основною задачею є знаходження ключа.

5.1.Принципи частотного крипто аналізу

Встановлено, що в будь-якій мові літери абетки зустрічаються нерівномірно. Якщо взяти достатньо великий текст (порядку мільйона символів) загального змісту та підрахувати частоту, з якою кожна літера абетки зустрічається в цьому тексті, ми побачимо, що найчастіше в українських текстах зустрічається літера „О" (0.082), а в російських та англійських – „Е" (0.071 та 0.12 відповідно). Звичайно, в залежності від тематики текстів, частотні характеристики його змінюються, але тенденція залишається незмінною. На цьому факті ґрунтується метод частотного криптоаналізу. Якщо метод шифрування „перехопленої шифровки" не приховує частотних особливостей мови (а саме таким і є шифр Цезаря), то криптоаналітики виконують наступні дії:

Підраховують відносні частоти, з якими кожна літера абетки зустрічається в „перехопленому” повідомленні. Робиться це за формулою: *частота=кількість/довжина*; де *кількість* – скільки разів літера зустрічається в повідомленні; *довжина* – кількість літер в повідомленні.

Літеру з найбільшою відносною частотою ототожнюють з літерою, яка має найбільшу частоту в таблиці.

Визначають величину зсуву.

Пробують дешифрувати повідомлення з визначеною в п.3 величиною зсуву. Якщо отримано логічний зв‘язний текст, повідомлення вважається дешифрованим. Якщо зв‘язного тексту не отримано, процедуру продовжують.

Літеру з найбільшою відносною частотою ототожнюють з літерою, яка має другу найбільшу частоту в таблиці.

Пробують дешифрувати повідомлення, перебираючи частотну таблицю, поки не отримують зв‘язного тексту.

Цим методом Вам необхідно користуватися для криптоаналізу в цій лабораторній роботі.

Криптографічна система RSA (Rivest, Shamir, Adleman), запропонована Рівестом, Шаміром і Едлеманом, належить до криптографічних систем з відкритим ключем. Її стійкість обумовлена великими проблемами при знаходженні розкладання великих простих чисел на множники.

Для того, щоби організувати передачу шифрованих повідомлень за допомогою криптосистеми RSA, необхідно зробити наступне:

За допомогою спеціальних алгоритмів згенерувати два великих простих числа p і q, які необхідно тримати у тайні.

Повідомити відправнику повідомлень (або розмістити у відкритому каталозі) число $n=pq$, а також випадкове ціле число E, взаємно просте з добутком $(p-1)(q-1)$.

Для розшифровки повідомлень, зашифрованих на відкритому ключі n, E, отримувачу необхідно мати число D, яке є мультиплікативним оберненим числа E за модулем $(p-1)(q-1)$, тобто $DE=1 \bmod (p-1)(q-1)$. Знайти таке число дуже просто, оскільки найбільший спільний дільник E і $(p-1)(q-1)$ якраз і рівний одиниці за вибором E.

Таким чином, відправник знає свій закритий ключ, n, E, а отримувач, крім того, знає ще свій секретний ключ D.

Довільне відкрите повідомлення можна уявити у вигляді послідовності цілих чисел з деякого інтервалу. Будемо вважати, що відправник передає секретне повідомлення у вигляді $X_1,\ldots,X_n$ $0<X_i<n-1$, для всіх i від 1 до k.

Відправник для кожного блоку X_i вираховує

$$C_i=(X_i^E) \bmod n \quad (1)$$

і передає C_i відкритим каналом зв'язку.

Маючи n, E і C_i, отримувач може розшифрувати повідомлення, використовуючи співвідношення

$X_i=(C_i^D) \bmod n$. (2)

Розглянемо в якості прикладу випадок p=3, q=11, n=3x11=33, E=7, D=3. Легко переконатися, що кожне з чисел E=7 і DE=21 взаємно прості з (p-1)(q-1)=20. Для передачі повідомлення М="02" відправнику треба обчислити $C=(2^7) \bmod 33=29$. Отримувач може розшифрувати повідомлення за допомогою такої операції: $X=29^3 \bmod 33=2$. =2. Якщо ж ми маємо текстове повідомлення, алфавіт якого пронумеровано від 00 до 32 (з пробілом), тоді можна зашифрувати довільне повідомлення російською мовою. Наприклад, якщо ми маємо повідомлення „ПРОВІРИМ ЗНАННЯ МАТЕМАТИКИ", то у зашифрованому вигляді на ключі n=33, E=7 воно буде мати вигляд:

27 25 20 29 14 25 02 12 32 28 07 00 07 02 14 32 00 25 02 26 12 14 06 02 10 02

Зрозуміло, що шифром в даному випадку є шифр простої заміни за табл. 1.

Таблиця 1. Таблиця заміни при шифруванні.

А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**Й**	**К**	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**
00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Т	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**	**Ю**	**Я**				
18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**			

Одним з відомих алгоритмів дешифрування системи RSA є метод ітерацій. Згідно з ним вихідне повідомлення можна отримати з шифрованого повторним шифруванням доти поки не отримаємо відкритий текст.

Приклад 1. Нехай p=383, q= 563, n=215629, E=49. В цьому випадку відкритий текст повністю отримується уже через 10 ітерацій повторного шифрування. Щоби в цьому впевнитися, достатньо довести, що $49^{10}=1 \bmod (p-1)(q-1)$. Виконання цієї рівності можна перевірити навіть на калькуляторі: (49^4=5764801 -> 49^4=183017 mod 214684 … 49^9=56957 mod 214684 -> 49^{10}= 1 mod 214684).

Інший метод атаки на шифр RSA – метод розкриття чисел p і q. Справа в тому, що $n=pq$ (як і самі ці числа p і q) повинні бути досить великими, щоби розкласти його на множники було дуже складно (в цьому і полягає складність цього алгоритму шифрування). Бажано, щоби p і q вибиралися випадковим чином і не були „дуже близькими" одне до одного. Покажемо, яким чином можна використати близькість значень p і q. Будемо вважати, що $p > q$ (що не накладає зайвих обмежень). Тоді для величин x=(p

+ q)/2, y=(p – q)/2 справедливе співвідношення: x^2-y^2=n. Перебираючи у порядку зростання варіанти x > $\sqrt{n}$, легко знайти розв'язок рівняння x^2-y^2=n, так як x=(p + q)/2 буде близьким до $\sqrt{n}$ у випадку близькості p і q.

Приклад 2. Нехай n=pq=851. Використаємо описаний спосіб для знаходження p і q. Так як $\sqrt{n}$ =29.17, беремо x=30 і обчислюємо 30^2-851=49 і з першої спроби знаходимо розв'язок x=30 і y=7. Таким чином, p=30+7=37, q=30-7=23.

Крім вказаних обмежень на p, q, E, D накладаються й інші обмеження.

Система шифрування RSA може бути застосована для цифрового підпису. У випадку підпису повідомлення M відправник обчислює $P=M^E$ mod n. Отримувач, який має M та P, перевіряє справедливість співвідношення $P^D=M$ mod n і впевнюється у справжності повідомлення M.

Приклад 3. Нехай p=3, q=11, n=3x11=33, E=7, D=3. Тоді відправник повідомлення M="02" обчислює цифровий підпис $P=2^7$ mod 33=29 і відправляє повідомлення „02, 29" отримувачу. Той, в свою чергу, перевіряє справжність повідомлення „02", обчисливши M=(29^3) mod 33=2.

Насправді підписують не саме повідомлення, а його т.зв. хеш-функцію. Спочатку оригінальне повідомлення обробляється деякою функцією, яка має таку властивість, що приймає на вході рядки різної довжини, а на виході видає деякий „дайджест", як правило, однакової і меншої, ніж вхідна, довжини. Хеш-функція виконує математичні обчислення, у результаті яких обчислюється значення хеш-функції. Хеш-функція може бути дуже простою. Наприклад, вона може виконати підсумовування всіх одиниць двійкового коду, або додати значення кодів всіх літер рядка, що обробляється (т.зв. контрольна сума) і т.д. Головне полягає в тому, що значення хеш-функції повинно залежати від усього вхідного рядка, щоби не можна було (в крайньому разі було б дуже важко) підібрати два різних вхідних рядки з однаковим значенням хеш-функції. Якщо таке трапляється, то кажуть що виникла колізія. Ми будемо користуватися найпростішою хеш-функцією, яка дуже недосконала і може викликати значні колізії. Однак, вона дуже проста і не потребує витрат машинного часу, а також складного програмування. Ця функція просто сумує всі значення символів за табл. 1 за модулем 33:

$$H(M)=\sum_{i=1,n} m_i \bmod 33. \quad (3)$$

До отриманого таким чином числа застосовують алгоритм прикладу 3, отримуючи, таким чином, зашифрований цифровий підпис. Отримувач, маючи повідомлення і цифровий підпис, розшифровує текст повідомлення, знаходить хеш-функцію від нього за формулою (3), розшифровує цифровий підпис, і порівнює отримані значення. Якщо вони однакові, повідомлення і цифровий підпис є істинними.

Проблема адміністрування криптографічними ключами вважається основним недоліком симетричних криптоалгоритмів. Цю проблему можна вирішити за допомогою асиметричної криптографії, тобто взагалі не використовувати симетричні криптоалгоритми. Однак такий підхід вважають нераціональним, оскільки асиметричні алгоритми працюють значно повільніше за симетричні і не можуть використовуватися у ряді важливих криптографічних застосувань. Іншим способом розповсюдження ключів є специфічні алгоритми, розроблені спеціально для таких застосувань. Одним з таких алгоритмів відкритого розповсюдження ключів є алгоритм Діффі-Хеллмана. Нехай учасники інформаційного обміну, сторони А і В, домовилися використати цей алгоритм для обміну ключами. Для цього необхідно виконати наступні обчислення. Спочатку А і В обирають велике просте число p, модуль системи. Для цього числа p обирають первісний корінь a. Числа p і a відкрито передають по каналах зв‘язку, так щоб їх мали обидві сторони.

Далі виконується наступний протокол:

А генерує ціле велике випадкове число x і відправляє В число:

$$X = a^x \bmod p \ ;$$

2) В генерує велике ціле випадкове число y і відправляє А число:

$$Y = a^y \bmod p \ ;$$

3) А обчислює:

$$k = Y^x \bmod p ;$$

4) В обчислює:

$$k' = X^y \bmod p \ .$$

І k, і k' дорівнюють $k = k' = a^{xy} \bmod p$.

Отже сторони А і В отримали один і той самий криптографічний ключ, не пересилаючи його каналами зв‘язку. Ніхто з осіб, що прослуховують цей канал, не зможе обчислити значення ключа. Адже їм відомі тільки p, a, X, Y, а для знаходження ключа необхідно розв‘язати задачу дискретного логарифмування. Тому А і В мають цілком таємний ключ, який більше ніхто не знає. Вибір a і p може помітно впливати на безпеку системи. Найголовніше, це те, що p повинно бути великим, таким, щоби задача дискретного логарифмування у скінченому полі була складною обчислювальною проблемою. Можна обирати довільне a, яке є первісним коренем за модулем p; немає причин, за якими не можна було б обрати a

найменшим з можливих, навіть однорозрядним. Навіть необов‘язково, щоби a було первісним коренем, воно повинно лише утворювати досить велику підгрупу мультиплікативної групи за модулем p. Програмний генератор двійкових послідовностей BBS (назву утворено від перших літер його авторів – Ленори та Мануеля Блум та Майка Шуба, Blum-Blum-Shub) вважають одним з найсильніших програмних генераторів псевдовипадкових послідовностей. Він вважається криптографічно стійким, і може використовуватися у серйозних криптографічних застосуваннях .

Нехай є два простих числа, p і q, причому $p \equiv q \equiv 3 \bmod 4$. Добуток цих чисел $n=pq$ називається цілим числом Блума. Оберемо ще одне випадкове число, x, взаємно просте з n та обчислимо $x_0 \equiv x \bmod n$. Це число вважається стартовим числом генератора. Далі можна обчислити наступні біти послідовності за формулою: $x_i \equiv x_{i-1}^2 \bmod n$ та $s_i \equiv x_i \bmod 2$. Останнє визначає, що в якості виходу генератора обирається молодший біт числа x_i . Найцікавішою властивістю генератора BBS є те, що для визначення значення i-го біту зовсім необов‘язково знати усі попередні i-1 бітів. Для безпосереднього обчислення значення i-го біту достатньо знати p та q. Безпека цієї схеми ґрунтується на складності розкладання n на множники. Число n можна опублікувати, так що кожен зможе генерувати біти за допомогою цього генератора. Однак поки криптоаналітик не розкладе n на множники, він не зможе передбачити вихід генератора. Більше того, генератор BBS непередбачуваний як в правому, так і в лівому напрямках. Це означає, що отримавши послідовність бітів, криптоаналітик не зможе передбачити ні наступний, ні попередній біти послідовності. Причиною цього є не якійсь заплутаний механізм генерації, а математика розкладання n на множники.

Приклад:

$p=19; q=23$

I	0	1	2	3	4	5	6	7
x_i	101	150	213	358	123	271	25	188
S_i	1	0	1	0	1	1	1	0

$p=q \equiv 3 \bmod 4$

$n=437$

$x=233$

Обов’язковою умовою, що накладається на зародок x, повинно бути наступне:

а) x – просте; б) x не ділиться на p і на q.

Цей генератор повільний, але є спосіб його прискорити. Як вказано у [2], в якості бітів псевдовипадкової послідовності можна використовувати не один молодший біт, а $\log_2 m$ молодших бітів, де m – довжина числа x_i. Порівняна повільність цього генератора не дозволяє використовувати його

для потокового шифрування (цей недолік зі зростанням швидкодії комп‘ютерів стає менш актуальним), а от для високонадійних застосувань, як наприклад, генерування ключів, він вважається кращим за багато інших.

Правильне функціонування підсистеми безпеки комп‘ютерної системи вимагає реалізації ряду функцій загального призначення, пов‘язаних з перетворенням вмісту об‘єктів системи (файлів, записів бази даних тощо) або з обчислення деяких спеціальних функцій, які суттєво залежать від вмісту об‘єктів. До таких функцій належать алгоритми контролю цілісності об‘єктів, аутентифікації та авторизації об‘єктів, що керують процесами, а також алгоритми підтримання конфіденційності інформації, що міститься в об‘єктах комп‘ютерної системи. Міжнародні та національні стандарти описують ряд добре відомих та вивчених функцій захисного характеру, зокрема алгоритми хешування MD5, MD2, SHA тощо; алгоритми генерування та перевірки електронного цифрового підпису RSA, DSS та інших. Усі ці алгоритми мають різні механізми викликів (зокрема, різну довжину аргументів). Це, у свою чергу, означає, що вони несумісні між собою. Тому задача вбудовування тих чи інших захисних механізмів в операційну систему на основі якогось одного алгоритму буде виглядати неефективною, особливо, якщо ця ОС розповсюджується в різних регіонах земної кулі. В цьому випадку логічним є побудова «шаруватої» структури, де окремий шар, реалізований, скажемо, як набір динамічних бібліотек, відповідає за захист інформації. Цей спосіб досить універсальний і широко застосовується у сімействі операційних систем Windows. Таким способом можна розв‘язати великий клас задач, пов‘язаних з універсалізацією ОС: від національних налаштувань системи до реалізації різноманітних засобів безпеки. Зрозуміло, що такі структури повинні мати т.зв. «відкритий інтерфейс», тобто бути детально документованими для того, щоби програмісти могли використати засоби цієї структури при створенні прикладного програмного забезпечення, в тому числі і для захисту інформації. Сьогодні є достатня кількість криптографічних інтерфейсів, однак найбільшої популярності набув інтерфейс від Microsoft - Microsoft CryptoAPI. Зараз використовується CryptoAPI версії 2.0. Причина популярності цього інтерфейсу полягає в тому, що Microsoft інтенсивно впровадила захисні механізми CryptoAPI у свої операційні системи та прикладне програмне забезпечення. Сучасні ОС сімейства Windows містять багато криптографічних підсистем різного призначення як прикладного рівня, так і рівня ядра. Провідну роль в цьому грають якраз функції CryptoAPI, зокрема базові криптографічні функції, сукупність яких створює інтерфейс CryptoAPI 1.0.

Інтерфейс CryptoAPI 2.0 містить як базові криптографічні функції, так і функції, що реалізують перетворення вищого рівня – роботу з сертифікатами X.509, обробку криптографічних повідомлень PKCS#7 та інші функції, що підтримують інфраструктуру відкритих ключів. Однак набір базових криптографічних функцій цього інтерфейсу утворює CryptoAPI 1.0.

Таким чином, функції CryptoAPI 1.0 утворюють криптографічне ядро прикладного рівня для сучасних операційних систем лінійки Windows. Програміст, який працює з цим інтерфейсом, може отримати усю необхідну інформацію про певного криптопровайдера засобами функції *CryptGetProvParam*. Перше, що необхідно знати при цьому – це набір криптографічних стандартів, які реалізують встановлені у системі криптопровайдери. Окрім різниці у стандартах, криптопровайдери відрізняються способом фізичної організації збереження ключової інформації. З точки зору програмування спосіб зберігання ключів значення не має, однак він дуже важливий з точки зору експлуатації та безпеки комп‘ютерної системи. Існуючі криптопровайдери Microsoft зберігають ключову інформацію на жорсткому диску (у реєстрі або у файлах), а провайдери інших фірм (GemPlus, Schlumberger та Infineon) – на смарт-картках. Якщо способи фізичної організації збереження ключової інформації у криптопровайдерів відрізняється, то логічна структура, яка визначається інтерфейсами та з якою мають справу програмісти, однакова для будь-якого типу провайдера. Ключова база визначається набором ключових контейнерів, кожен з яких має ім‘я, що привласнюється йому при створенні, а потім використовується для роботи з ним. У ключовому контейнері зберігається довготривала ключова інформація, наприклад, ключові пари для цифрового підпису або несиметричної системи шифрування.

Тепер розглянемо детально, як функції інтерфейсу CryptoAPI викликають бібліотеки конкретного криптопровайдера. Кожен криптопровайдер має своє власне ім‘я та тип. Його ім‘я – просто рядок, за допомогою якого система його ідентифікує. Так, базовий криптопровайдер Microsoft має назву Microsoft Base Cryptographic Provider v1.0. Тип криптопровайдера – ціле число (у нотації C – DWORD), значення якого ідентифікує набір криптографічних алгоритмів, що підтримуються. Криптопровайдер Microsoft має тип 1, цей тип провайдера реалізує в якості алгоритмів цифрового підпису та обміну ключів алгоритм RSA. Інший базовий криптопровайдер Microsoft, „Microsoft Base DSS and Diffie-Hellman Cryptographic Provider”, має тип 13. Цей тип криптопровайдера реалізує алгоритм цифрового підпису DSS, а в якості алгоритму обміну ключами – протокол Діффі-Хелмана.

Отже, для роботи з набором криптопровайдерами у системному реєстрі міститься список імен усіх криптопровайдерів. З кожним ім‘ям пов‘язаний тип криптопровайдера та ім‘я бібліотеки, яка реалізує його алгоритми.

Окрім цього в системі міститься інформація про те, який криптопровайдер треба застосовувати, якщо користувач явно не вказав конкретне його ім‘я, лише визначивши тип провайдера. Такий криптопровайдер називають провайдером за замовчуванням для заданого типу. Наприклад, для типу 1 провайдером за замовчуванням є Microsoft Base Cryptographic Provider v1.0, а для типу 13 - Microsoft Base DSS and Diffie-

Hellman Cryptographic Provider. Для визначення криптопровайдерів за замовчуванням використовують функцію *CryptGetDefaultProvider,* а для зміни цього параметру – функції *CryptSetProvider* або *CryptSetProviderEx*. Функції дозволяють встановити провайдера за замовчуванням як для поточного користувача, так і для системи в цілому (усіх користувачів). Ці параметри зберігаються у вулику реєстру HKEY_LOCAL_MACHINE. Параметри, встановлені для поточного користувача, мають пріоритет над параметрами, встановленими для усієї системи, та зберігаються у вулику реєстру HKEY_CURRENT_USER. Якщо параметри для поточного користувача відсутні, застосовуються загальносистемні.

Тепер розглянемо, яким чином користувач починає працювати з конкретним криптопровайдером, і як система викликає конкретну бібліотеку, що відповідає обраному криптопровайдеру.

Робота з певним провайдером починається з виклику функції *CryptAcquireContext*, де користувач визначає тип потрібного криптопровайдера, його назву та назву робочого ключового контейнера. В результаті роботи функція повертає користувачу дескриптор криптопровайдера (handle), за допомогою якого користувач в подальшому буде звертатися до нього та передавати його у процедури для виконання усіх необхідних криптографічних операцій. Детальний опис контексту роботи з криптопровайдерами та приклади (мовою програмування *C*) дивіться у книжці Щербакова Л.Ю., Домашева А.В. «Прикладная криптография».

Власне бібліотеки CryptoAPI разом з файлами заголовків та допомоги постачаються у складі бібліотек MSDN.

6. Криптографічні ключі

В основу шифрування покладено два елементи: криптографічний алгоритм і ключ.

Криптографічний алгоритм - це математична функція, яка комбінує відповідний текст або іншу зрозумілу інформацію з ланцюжком чисел (ключем) з метою отримання незв язаного (шифрованого) тексту.

Усі криптографічні алгоритми можна поділити на дві групи: загальні і спеціальні.

Спеціальні криптоалгоритми мають таємний алгоритм шифрування, а загальні криптоалгоритми характерні повністю відкритим алгоритмом, і їх криптостійкість визначається ключами шифрування. Спеціальні алгоритми найчастіше використовують в апаратних засобах криптозахисту.

Загальні криптографічні алгоритми часто стають стандартами шифрування, якщо їхня висока криптостійкість доведена. Ці алгоритми оприлюднюють для обговорення, при цьому навіть визначається премію за успішну спробу його "злому". Криптостійкість загальних алгоритмів визначається ключем шифрування, який генерується методом випадкових

чисел і не може бути повторений протягом певного часу. Криптостійкість таких алгоритмів буде вищою відповідно до збільшення довжини ключа.

Є дві великі групи загальних криптоа лгоритмі в: симетричні і асиметричні. До симетричних криптографічних алгоритмів належать такі алгоритми, для яких шифрування і розшифрування виконується однаковим ключем, тобто і відправник, і отримувач повідомлення мають користуватися тим самим ключем. Такі алгоритми мають досить велику швидкість обробки як для апаратної, так і для програмної реалізації. Основним їх недоліком є труднощі, пов'язані з дотриманням безпечного розподілу ключів між абонентами системи. Для асиметричних криптоалгоритмів шифрування і розшифрування виконують за допомогою різних ключів, тобто, маючи один із ключів, не можна визначити парний для нього ключ. Такі алгоритми часто потребують значно довшого часу для обчислення, але не створюють труднощів під час розподілу ключів, оскільки відкритий розподіл одного з ключів не зменшує криптостій-кості алгоритму і не дає можливості відновлення парного йому ключа.

Усі криптографічні алгоритми можна використовувати з різними цілями, зокрема:

• для шифрування інформації, тобто приховування змісту повідомлень і даних;

• для забезпечення захисту даних і повідомлень від модифікації.

З найпоширеніших методів шифрування можна виділити американський алгоритм шифрування DES (Data Encryption Standart, розроблений фахівцями фірми IBM і затверджений урядом США 1977 року) із довжиною ключа, що може змінюватися, та алгоритм ГОСТ 28147-89, який був розроблений та набув широкого застосування в колишньому СРСР і має ключ постійної довжини. Ці алгоритми належать до симетричних алгоритмів шифрування.

Алгоритм Потрійний DES був запропонований як альтернатива DES і призначений для триразового шифрування даних трьома різними закритими ключами для підвищення ступеня захисту.

RC2, RC4, RC5 - шифри зі змінною довжиною ключа для дуже швидкого шифрування великих обсягів інформації. Здатні підвищувати ступінь захисту через вибір довшого ключа.

IDEA (International Data Encryption Algoritm) призначений для швидкої роботи в програмній реалізації.

Для приховування інформації можна використовувати деякі асиметричні алгоритми, наприклад, алгоритм RSA. Алгоритм підтримує змінну довжину ключа та змінний розмір блоку тексту, що шифрується.

Алгоритм RSA дозволяє виконувати шифрування в різних режимах:

• за допомогою таємного ключа відправника. Тоді всі, хто має його відкритий ключ, можуть розшифрувати це повідомлення;

• за допомогою відкритого ключа отримувача, тоді тільки власник таємного ключа, який є парним до цього відкритого, може розшифрувати таке повідомлення;

• за допомогою таємного ключа відправника і відкритого ключа отримувача повідомлення. Тоді тільки цей отримувач може розшифрувати таке повідомлення.

Але не всі асиметричні алгоритми дозволяють виконувати шифрування даних у таких режимах. Це визначається математичними функціями, які закладені в основу алгоритмів.

Другою метою використання криптографічних методів є захист інформації від модифікації, викривлення або підробки. Цього можна досягнути без шифрування повідомлень, тобто повідомлення залишається відкритим, незашифрованим, але до нього додається інформацію, перевірка якої за допомогою спеціальних алгоритмів може однозначно довести, що ця інформація не була змінена. Для симетричних алгоритмів шифрування така додаткова інформація - це код автентифікації, який формується за наявності ключа шифрування за допомогою криптографічних алгоритмів.

Для асиметричних криптографічних алгоритмів формують додаткову інформацію, яка має назву електронний цифровий підпис. Формуючи електронний цифровий підпис, виконують такі операції:

• за допомогою односторонньої хеш-функції обчислюють прообраз цифрового підпису, аналог контрольної суми повідомлення;

• отримане значення хеш-функції шифрується: а) таємним або відкритим; б) таємним і відкритим ключами відправника і отримувача повідомлення - для алгоритму RSA

• використовуючи значення хеш-функції і таємного ключа, за допомогою спеціального алгоритму обчислюють значення цифрового підпису, - наприклад, для російського стандарту Р.31-10.

Для того, щоб перевірити цифровий підпис, потрібно:

• виходячи із значення цифрового підпису та використовуючи відповідні ключі, обчислити значення хеш-функції;

• обчислити хеш-функцію з тексту повідомлення;

• порівняти ці значення. Якщо вони збігаються, то повідомлення не було модифікованим і відправлене саме цим відправником.

Останнім часом використання електронного цифрового підпису значно поширюється, у тому числі для регулювання доступу до конфіденційної банківської інформації та ресурсів системи, особливо для on-line -систем реального часу.

Ефективність захисту систем за допомогою будь-яких криптографічних алгоритмів значною мірою залежить від безпечного розподілу ключів. Тут можна виділити такі основні методи розподілу ключів між учасниками системи.

1. Метод базових/сеансових ключів. Такий метод описаний у стандарті ISO 8532 і використовується для розподілу ключів симетричних

алгоритмів шифрування. Для розподілу ключів вводиться ієрархія ключів: головний ключ (так званий майстер-ключ, або ключ шифрування ключів) і ключ шифрування даних (тобто сеансовий ключ). Ієрархія може бути і дворівневою: ключ шифрування ключів/ключ шифрування даних. Старший ключ у цій ієрархії треба розповсюджувати неелектронним способом, який виключає можливість його компрометації. Використання такої схеми розподілу ключів потребує значного часу і значних затрат.

2. Метод відкритих ключів. Такий метод описаний у стандарті ISO 11166 і може бути використаний для розподілу ключів як для симетричного, так і для асиметричного шифрування. За його допомогою можна також забезпечити надійне функціонування центрів сертифікації ключів для електронного цифрового підпису на базі асиметричних алгоритмів та розподіл сертифікатів відкритих ключів учасників інформаційних систем. Крім того, використання методу відкритих ключів дає можливість кожне повідомлення шифрувати окремим ключем симетричного алгоритму та передавати цей ключ із самим повідомленням у зашифрованій асиметричним алгоритмом формі.

Вибір того чи іншого методу залежить від структури системи і технології обробки даних. Жоден із цих методів не забезпечує "абсолютного" захисту інформації, але гарантує, що вартість "злому" у кілька разів перевищує вартість зашифрованої інформації.

Щоб використовувати систему криптографії з відкритим ключем, потрібно генерувати відкритий і особистий ключі. Після генерування ключової пари слід розповсюдити відкритий ключ респондентам. Найнадійніший спосіб розповсюдження відкритих ключів - через сертифікаційні центри, що призначені для зберігання цифрових сертифікатів.

Цифровий сертифікат - це електронний ідентифікатор, що підтверджує справжність особи користувача, містить певну інформацію про нього, слугує електронним підтвердженням відкритих ключів.

Сертифікаційні центри несуть відповідальність за перевірку особистості користувача, надання цифрових сертифікатів та перевірку їхньої справжності.

7.Криптографічні протоколи

Протокол - це послідовність кроків, які вживають дві або більша кількість сторін для спільного вирішення деякої задачі. Слід звернути увагу на те, що всі кроки робляться в порядку суворої черговості і жоден з них не може бути зроблений раніше, ніж закінчиться попередній. Крім того, будь-який протокол передбачає участь двох сторін. Поодинці можна змішати і випити коктейль, але до протоколу ці дії не будуть мати ніякого відношення. Тому доведеться почастувати когось зробленою коктейлем, щоб його приготування

і дегустація стали справжнім протоколом. І нарешті, протокол обов'язково призначений для досягнення якоїсь мети, інакше це не протокол, а пусте проведення часу. У протоколів є також і інші відмітні риси:

• кожен учасник протоколу повинен бути заздалегідь сповіщений про кроки, які йому належить зробити;

• всі учасники протоколу повинні слідувати його правилам добровільно, без примусу;

• необхідно, щоб протокол допускав лише однозначне тлумачення, а його кроки були абсолютно чітко визначені і не допускали можливості їх неправильного розуміння;

• протокол повинен описувати реакцію учасників на будь-які ситуації, які можуть виникнути в ході його реалізації. Іншими словами, неприпустимим є положення, при якому для ситуації, що виникла протоколом не визначено відповідне действие. Криптографічним протоколом називається протокол, в основі якого лежить криптографічний алгоритм. Проте метою криптографічного протоколу часто є не тільки збереження інформації в таємниці від сторонніх. Учасники криптографічного протоколу можуть бути близькими друзями, у яких немає один від одного секретів, а можуть бути і непримиренними ворогами, кожен з яких відмовляється повідомити іншому, яке сьогодні число. Проте їм може знадобитися поставити свої підписи під спільним договором або засвідчити свою особистість. У цьому випадку криптографія потрібна, щоб запобігти або виявити підслуховування сторонніми особами, а також не допустити шахрайства. Тому часто криптографічний протокол потрібно там, де його учасники не повинні зробити або дізнатися більше того, що визначено цим протоколом. Навіщо потрібні криптографічні протоколи У повсякденному житті нам доводиться стикатися з протоколами буквально на кожному кроці - граючи в будь-які ігри, або роблячи покупки в магазинах, або голосуючи на виборах. Багатьма протоколами нас навчили користуватися батьки, шкільні вчителі та друзі. Решта ми зуміли дізнатися самостійно.
В даний час люди все частіше контактують один з одним за допомогою комп'ютерів. Комп'ютери ж, на відміну від більшості людей, в школу не ходили, у них не було батьків, та й вчитися без допомоги людини вони не в змозі. Тому комп'ютери доводиться постачати формалізованими протоколами, щоб вони змогли робити те, що люди виконують не замислюючись. Наприклад, якщо в магазині не виявиться касового апарату, ви все одно опинитеся в змозі купити в ньому необхідну для себе річ. Комп'ютер ж таке кардинальна зміна протоколу може поставити у глухий кут. Більшість протоколів, які люди використовують при спілкуванні один з одним віч-на-віч, добре себе зарекомендували тільки тому, що їх учасники мають можливість вступити в безпосередній контакт. Взаємодія з іншими людьми через комп'ютерну мережу, навпаки, передбачає анонімність. Чи будете ви грати з незнайомцем в преферанс, не бачачи, як він тасує колоду і роздає карти? Довірите ви свої гроші зовсім сторонній людині, щоб він купив

вам що-небудь в магазині? Пошлете ви свій бюлетень голосування пошті, знаючи, що з ним зможе ознайомитися хтось із поштових працівників і потім розповісти всім про ваші нетрадиційних політичних уподобаннях? Думаю, що ні. Нерозумно вважати, що комп'ютерні користувачі поводяться чесніше, ніж абсолютно випадкові люди. Те ж саме стосується і мережевих адміністраторів, і проектувальників комп'ютерних мереж. Більшість з них і справді досить чесні, проте інші можуть заподіяти вам дуже великі неприємності. Тому так потрібні криптографічні протоколи, використання яких дозволяє захиститися від непорядних людей. Розподіл ролей Щоб опис протоколів була більш наочним, їх учасники будуть носити імена, які однозначно визначають ролі, їм уготовані (див. таблицю). Антон і Борис беруть участь у всіх двосторонніх протоколах. Як правило, починає виконання кроків, передбачених протоколом, Антон, а відповідні дії вживає Борис. Якщо протокол є трьох-або чотирістороннім, виконання відповідних ролей беруть на себе Володимир та Георгій. Про решту персонажів докладніше йтиметься дещо пізніше. Протоколи з арбітражем Арбітр є незацікавленим учасником протоколу, якому решта учасників повністю довіряють, роблячи відповідні дії для завершення чергового кроку протоколу. Це означає, що у арбітра немає особистої зацікавленості у досягненні тих чи інших цілей, переслідуваних учасниками протоколу, і він не може виступити на боці одного з них. Учасники протоколу також приймають на віру все, що скаже арбітр, і беззаперечно виконують всім його рекомендаціям.

У протоколах, яких ми дотримуємося у повсякденному житті, роль арбітра найчастіше грає адвокат. Однак спроби перенести протоколи з адвокатом як арбітра з повсякденні в комп'ютерні мережі наштовхуються на суттєві перешкоди:

• Легко довіритися адвокату, про якого відомо, що в нього незаплямована репутація і з яким можна встановити особистий контакт. Але якщо два учасника протоколу не довіряють один одному, арбітр, не одягнений у тілесну оболонку і існуючий десь у надрах комп'ютерної мережі, навряд чи буде користуватися в них великою довірою.

• Розцінки на послуги, що надаються адвокатом, відомі. Хто і яким чином буде оплачувати аналогічні послуги арбітра у комп'ютерній мережі?

• Введення арбітра в будь-який протокол збільшує час, що витрачається на реалізацію цього протоколу.

• Оскільки арбітр контролює кожен крок протоколу, його участь у дуже складних протоколах може стати вузьким місцем при реалізації таких протоколів. Відповідне збільшення числа арбітрів дозволяє позбутися даного вузького місця, проте одночасно збільшуються і витрати на реалізацію протоколу.

• У силу того, що всі учасники протоколу повинні користуватися послугами одного й того ж арбітра, дії зловмисника, який вирішить завдати їм шкоди, будуть спрямовані, в першу чергу, проти цього арбітра. Отже, арбітр є слабка

ланка в ланцюзі учасників будь-якого протоколу з арбітражем. Незважаючи на зазначені перешкоди, протоколи з арбітражем знаходять широке застосування на практиці. Протокол із суддівством Щоб знизити накладні витрати на арбітраж, протокол, в якому бере участь арбітр, часто ділиться на дві частини. Перша повністю збігається зі звичайним протоколом без арбітражу, а до другої вдаються лише в разі виникнення розбіжностей між учасниками. Для вирішення конфліктів між ними використовується особливий тип арбітра - суддя. Подібно арбітру, суддя є незацікавленим учасником протоколу, якому інші його учасники довіряють при прийнятті рішень. Однак на відміну від арбітра, суддя бере участь аж ніяк не в кожному кроці протоколу. Послугами судді користуються, тільки якщо потрібно дозволити сумніви щодо правильності дій учасників протоколу. Якщо таких сумнівів ні у кого не виникає, суддівство не знадобиться. У комп'ютерних протоколах з суддівством передбачається наявність даних, перевіривши які довірена третя особа може вирішити, не змахлював чи хто-небудь з учасників цього протоколу. Хороший протокол з суддівством також дозволяє з'ясувати, хто саме веде себе нечесно. Це служить прекрасним превентивним засобом проти шахрайства з боку учасників такого протоколу. Самостверджуються протокол Самостверджуються протокол не вимагає присутності арбітра для завершення кожного кроку протоколу. Він також не передбачає наявність судді до розв'язання конфліктних ситуацій. Самостверджуються протокол влаштований так, що якщо один з його учасників шахраювати, інші зможуть моментально розпізнати нечесність, виявлену цим учасником, і припинити виконання наступних кроків протоколу. Звичайно ж, хочеться, щоб існував універсальний самостверджуються протокол на всі випадки життя. Однак на практиці в кожному конкретному випадку доводиться конструювати свій спеціальний самостверджуються протокол. Різновиди атак на протоколи Атаки на протоколи бувають спрямовані проти криптографічних алгоритмів, які в них задіяні, проти криптографічних методів, що застосовуються для їх реалізації, а також проти самих протоколів. Для початку припустимо, що використовуються криптографічні алгоритми та методи є досить стійкими, і розглянемо атаки власне на протоколи. Особа, яка не є учасником протоколу, може спробувати підслухати інформацію, якою обмінюються його учасники. Це пасивна атака на протокол, яка названа так тому, що атакуючий (будемо іменувати його Петром) може тільки накопичувати дані і спостерігати за ходом подій, але не в змозі впливати на нього. Пасивна атака подібна криптоаналітичних атаці зі знанням тільки шифртекста. Оскільки учасники протоколу не володіють надійними засобами, що дозволяють їм визначити, що вони стали об'єктом пасивної атаки, для захисту від неї використовуються протоколи, що дають можливість запобігати можливі несприятливі наслідки пасивної атаки, а не розпізнавати її. Атакуючий може спробувати внести зміни в протокол заради власної вигоди. Він може видати себе за учасника

протоколу, внести зміни до повідомлення, якими обмінюються учасники протоколу, підмінити інформацію, яка зберігається в комп'ютері і використовується учасниками протоколу для прийняття рішень. Це активна атака на протокол, оскільки атакуючий (назвемо його Зіновієм) може втручатися в процес виконання кроків протоколу його учасниками. Отже, Петро намагається зібрати максимум інформації про учасників протоколу і про їхні дії. У Зиновія ж зовсім інші інтереси - погіршення продуктивності комп'ютерної мережі, отримання несанкціонованого доступу до її ресурсів, внесення спотворень у бази даних. При цьому і Петро, і Зіновій не обов'язково є абсолютно сторонніми особами. Серед них можуть бути легальні користувачі, системні і мережні адміністратори, розробники програмного забезпечення і навіть учасники протоколу, які поводяться непорядно і навіть зовсім не дотримуються цей протокол. В останньому випадку атакуючий називається шахраєм. Пасивний шахрай слід усім правилам, які визначені протоколом, але при цьому ще й намагається дізнатися про інших учасників більше, ніж передбачено цим протоколом. Активний шахрай вносить довільні зміни в протокол, щоб нечесним шляхом домогтися собі найбільшої вигоди. Захист протоколу від дій кількох активних шахраїв є дуже нетривіальну проблему. Проте за деяких умов цю проблему вдається вирішити, надавши учасникам протоколу можливість вчасно розпізнати ознаки активного шахрайства. А захист від пасивного шахрайства повинен надавати будь-який протокол незалежно від умов, в які поставлені його учасники.

8.Надійність криптосистем

8.1.Як вибрати хороший криптографічний алгоритм

Безпека криптосистем можна порівняти з надійністю ланцюга: чим міцніше її найслабша ланка, тим важче порвати цей ланцюг. У хорошій криптосистема має бути досконально перевірено всі - алгоритм, протокол. ключі та т. п. Якщо криптографічний алгоритм досить стійкий, а генератор випадкових чисел, який використовується для породження ключів, нікуди не годиться, будь-який досвідчений криптоаналітика в першу чергу зверне увагу саме на нього. Якщо вдасться поліпшити генератор, але не будуть зачищені комірки пам'яті комп'ютера після того, як у них побував згенерований ключ, гріш ціна такої безпеки. Розглянемо таку ситуацію. У криптосистема застосовуються стійкий криптографічний алгоритм і дійсно випадкові ключі, які акуратно видаляються з пам'яті комп'ютера після їх використання. Однак перед шифруванням файл, в якому поряд з вашим адресою та прізвищем зазначені всі ваші доходи за поточний рік, був помилково відправлений електронною поштою в податкову службу. У цьому випадку можна запитати. навіщо тоді вам знадобилися і стійкий алгоритм, і випадкові ключі, і зачистка комп'ютерної пам'яті на додачу?! Криптограф не позаздриш: у проектованої їм криптосистема він повинен передбачити захист від атак усіх типів, які

тільки зможе придумати запалене уява криптоаналітика. Криптоаналітика ж навпаки досить відшукати єдине слабка ланка в ланцюзі криптографічного захисту та організувати атаку проти цього дзвени. Крім цього, завжди слід враховувати, що на практиці загроза інформаційної безпеки будь-якого об'єкта виходить не тільки від криптоаналітика. Зрештою, яким би довгим не був криптографічний ключ, використовуваний вами для шифрування файлів, все одно, якщо правоохоронним органам знадобиться дізнатися, що зберігається у вашому комп'ютері. вони просто встановлять камеру і скрупульозно запишуть всю інформацію, що з'являється на його екрані. Недарма, за визнанням офіційних лип з АНБ, більшість збоїв у забезпеченні інформаційної безпеки відбувається не через знайдені слабкостей в криптографічних алгоритмах і протоколах, а з-за кричущих помилок при їх реалізації. Який би стійкістю не володів криптографічний алгоритм, її не потрібно долати в лоб, т. к. при успішній атаці її вдається просто обійти. Однак і нехтувати хорошими криптографічними алгоритмами теж не слід, щоб криптографія не стала найслабшою ланкою в ланцюзі, яке не витримає натиску атакуючого. При виборі хорошого криптографічного алгоритму можна: скористатися відомим алгоритмом, порівняно давно опублікованим в спеціальному виданні, присвяченому проблемам криптографії (якщо ніхто поки не повідомив про те, що зумів розкрити цей алгоритм. значить на нього слід звернути увагу); довіритися відомій фірмі, що спеціалізується на продажу засобів шифрування (навряд чи ця фірма буде ризикувати своїм добрим ім'ям. торгуючи нестійкими криптографічними алгоритмами); звернутися до незалежного експерта (неупередженість в думці дозволить йому об'єктивно оцінити достоїнства і недоліки різних криптографічних алгоритмів); звернутися за підтримкою до відповідного урядове відомство (навряд чи уряд буде вводити своїх громадян в оману. даючи їм помилкові поради щодо стійкості того пли іншого криптографічного алгоритму); спробувати створити власний криптографічний алгоритм.
Всі перераховані варіанти мають суттєві вади. Не слід покладатися тільки на одну фірму, на одного експерта або на одне відомство. Багато людей, які називають себе незалежними експертами, мало розуміють в криптографії. Більшість фірм, що виробляють засоби шифрування, - теж нітрохи не краще. У АНБ і ФАПСИ працюють найкращі криптографи в світі, однак зі зрозумілих міркувань вони не поспішають поділитися своїми секретами з першим зустрічним. Втім, і з другим теж. І навіть якщо ви геній у галузі криптографії, нерозумно використовувати криптографічний алгоритм власного винаходу без того, щоб його всебічно не проаналізували і не протестували досвідчені криптологи. Тому найбільш кращою представляється перша з перерахованих можливостей. Даний підхід до оцінки стійкості криптографічних алгоритмів можна було б визнати ідеальним, якби не один його не-достаток. На жаль, нічого не відомо про результати криптоаналітичних досліджень цих алгоритмів, які безсумнівно активно велися в. Минулому і продовжують також активно проводитися в

усьому світі численними співробітниками різних урядових відомств, до компетенції яких входять Криптологічних вишукування. Ці відомства, швидше, всього, набагато краще фінансуються, ніж академічні інститути, провідні аналогічні дослідження. Та й почали вони займатися криптології значно раніше, ніж вчені, не мають військових звань, і фахівці з приватних фірм. Тому можна припустити, що військові знайшли набагато простіші способи розкриття відомих шифрів, ніж ті, які винайдені за межами суворо охоронюваних будівель надсекретних урядових відомств.
Ну і нехай. Навіть якщо вас заарештують і як доказ конфіскують у вас жорсткий диск з файлами, зашифрованими по DES-алгоритму. навряд чи криптоаналитики, що складаються на державній службі, прийдуть на судове засідання, щоб клятвено підтвердити, що дані для вашого обвинувального висновку отримані шляхом дешифрування конфіскованих файлів. Той факт, що можна розкрити якийсь конкретний криптографічний алгоритм, часто є значно більшим секретом, ніж інформація, отримана шляхом розкриття цього алгоритму. Краще виходити з припущення, що АНБ, ФАПСИ і іже з ними можуть, прочитати будь-яке повідомлення, яке вони побажають прочитати. Однак ці відомства не в змозі читати всі повідомлення, з вмістом яких хочуть ознайомитися. Головною причиною є обмеженість у коштах, що асигнуються урядом на криптоаналіз. Інше розумне припущення полягає в тому, що компетентним органам набагато легше отримати доступ до зашифрованої інформації за допомогою грубої фізичної сили, ніж шляхом витончених, але дуже трудомістких математичних викладок, що призводять до розкриття шифру. Однак у кожному випадку набагато надійніше користуватися відомим криптографічним алгоритмом, який придуманий вже досить давно і зумів вистояти проти численних спроб розкрити його, зроблених авторитетними криптології. Криптографічні алгоритми, призначені для експорту з США. В даний час у користувачів персональних комп'ютерів є можливість застосовувати алгоритми шифрування, вбудовані в різні програмні продукти. Досить придбати, наприклад, текстовий редактор Word, редактор електронних таблиць Excel або операційні системи Windows NT і NetWare. Крім вбудованих алгоритмів шифрування, всі ці програмні продукти мають ще одну спільну властивість: вони виготовлені в Сполучених Штатах. Перш ніж почати торгувати ними за кордоном, американські виробники в обов'язковому порядку повинні отримати дозвіл у свого уряду на експорт даних продуктів за межі США. Багато хто дотримується зараз такої думки: жоден криптографічний алгоритм, дозволений до експорту із США, не є достатньо стійким, щоб його не могли розкрити криптоаналитики з АНБ. Вважається, що американські компанії, які бажають продавати за кордоном свою продукцію, яка дозволяє шифрувати дані, за наполяганням АНБ переробляють використовуються криптографічні алгоритми так, що: час від часу окремі біти ключа підмішуються в шифртекст; ключ має довжину всього 30 біт замість офіційно заявлених 100 біт, оскільки більшість ключів виявляються еквівалентні; на початок кожного

шіфруемого повідомлення вставляється фіксований заголовок, щоб полегшити криптоаналітичних атаку зі знанням відкритого тексту; будь шифроване повідомлення містить деякий фрагмент відкритого тексту разом з відповідним йому шифртекста. Вихідні тексти американських шифрувальних програм передаються на зберігання в АНБ, однак за межами цього надсекретного агентства доступ до них закритий наглухо. Цілком природно, що ні АНБ, ні американські компанії, що отримали від АНБ дозвіл на експорт своїх шифрувальних засобів, не зацікавлені в рекламі слабкостей криптографічних алгорт-мов, покладених в основу функціонування цих коштів. Тому бажано проявляти обережність, якщо ви збираєтеся захищати свої дані за допомогою американських програм шифрування, експорт яких за межі країни дозволений урядом США.

8.2.Симетричний або асиметричний криптографічний алгоритм?

Який алгоритм краще - симетричний або асиметричний? Питання не цілком коректний, оскільки передбачає використання однакових критеріїв при порівнянні криптосистем з секретним і відкритим ключами. А таких критеріїв просто не існує. Тим не менш, дебати щодо переваг та недоліків двох основних видів криптосистем ведуться давно, починаючи з моменту винаходу першого алгоритму з відкритим ключем. Відзначено, що симетричні криптографічні алгоритми мають меншу довжину ключа і працюють швидше, ніж асиметричні. Однак, на думку американського криптолога У. Діффі - одного з винахідників і криптосистем з відкритим ключем-Їх слід розглядати не як абсолютно новий різновид універсальних криптосистем. Криптографія з відкритим ключем і криптографія з секретним ключем призначені для вирішення абсолютно різних проблем, пов'язаних з засекречуванням інформації. Симетричні криптографічні алгоритми служать для шифрування даних, вони працюють на кілька порядком швидше, ніж асиметричні алгоритми. Однак криптографія з відкритим ключем успішно використовується в таких областях, для яких криптографія з секретним ключем підходить погано, - наприклад, при роботі з ключами і з переважною більшістю криптографічних протоколів.

8.3.Шифрування в каналах зв'язку комп'ютерної мережі

Однією з відмінних характеристик будь-якої комп'ютерної мережі є її поділ на так звані рівні, кожен з яких відповідає за дотримання певних умов і виконання функцій, необхідних для спілкування між комп'ютерами, пов'язаними в мережу. Цей поділ на рівні має фундаментальне значення для створення стандартних комп'ютерних мереж. Тому в 1984 р. кілька міжнародних організацій і комі готовий об'єднали свої зусилля і виробили приблизну модель комп'ютерної мережі, відому під назвою OSI (Open.

Systems Interconnection - Модель відкритих мережевих з'єднань). Відповідно до моделі OSI комунікаційні функції рознесені по рівнях. Функції кожного рівня незалежні від функцій нижче-і вищих рівнів. Кожен рівень може безпосередньо спілкуватися тільки з двома сусідніми. Модель OSI визначає 7 рівнів: верхні 3 служать для зв'язку з кінцевим користувачем, а нижні 4 орієнтовані на виконання комунікаційних функцій у реальному масштабі часу. Теоретично шифрування даних для передачі по каналах зв'язку комп'ютерної мережі може здійснюватися на будь-якому рівні моделі OSI. На практиці "це зазвичай робиться або на самих нижніх, або на самих верхніх рівнях. Якщо дані шифруються на нижніх рівнях, шифрування називається канальним, а якщо на верхніх, то таке шифрування називається наскрізним. Обидва ці підходи до шифрування даних мають свої переваги і недоліки .Канальне шифрування При канальному шифруванні шифруються абсолютно всі дані, що проходять по кожному каналу зв'язку, включаючи відкритий текст повідомлення, а також інформацію про його маршрутизації і про використовуваний комунікаційному протоколі. Однак у цьому випадку будь-інтелектуальний мережевий вузол (наприклад, комутатор) буде змушений розшифровувати вхідний потік даних, щоб відповідним чином його обробити, знову зашифрувати і передати на інший вузол мережі.

Проте канальне шифрування являє собою дуже ефективний засіб захисту інформації в комп'ютерних мережах. Оскільки шифруванню підлягають всі дані, передані від одного вузла мережі до іншого, у криптоаналітика немає ніякої додаткової інформації про те, хто служить джерелом цих даних, кому вони призначені, яка їхня структура і т. д. А якщо ще подбати і про те, щоб , поки канал простоює, передавати по ньому випадкову бітову послідовність, сторонній спостерігач не зможе навіть сказати, де починається і де закінчується текст переданого повідомлення. Не надто складною є та робота з ключами. Однаковими ключами слід забезпечити тільки два сусідніх вузла мережі зв'язку, які потім можуть змінювати використовувані ключі незалежно від інших пар вузлів. Найбільший недолік канального шифрування полягає в тому, що дані доводиться шифрувати при передачі по кожній фізичній каналу комп'ютерної мережі. Відправлення інформації в незашифрованому вигляді по якомусь з каналів ставить під загрозу забезпечення безпеки всієї мережі. В результаті вартість реалізації канального шифрування у великих мережах може виявитися надмірно високою. Крім того, при використанні канального шифрування додатково буде потрібно захищати кожен вузол комп'ютерної мережі, по якому передаються дані. Якщо абоненти мережі повністю довіряють один одному і кожен її вузол розміщений там, де він захищений від зловмисників, на цей недолік канального шифрування можна не звертати уваги. Однак на практиці такий стан зустрічається надзвичайно рідко. Адже в кожній фірмі є конфіденційні дані, знайомитися з якими можуть тільки співробітники

одного певного відділу, а за його межами доступ до цих даних необхідно обмежувати до мінімуму.

Наскрізне шифрування

При наскрізному шифруванні криптографічний алгоритм реалізується на одному з верхніх рівнів моделі OSI. Шифруванню підлягає тільки змістовна частина повідомлення, яке потрібно передати по мережі. Після зашифрування до неї додається службова інформація, необхідна для маршрутизації повідомлення і результат переправляється на більш низькі рівні з метою відправки адресату. Тепер повідомлення не потрібно постійно розшифровувати і зашифровувати при проходженні через кожен проміжний вузол мережі зв'язку. Повідомлення залишається зашифрованим на всьому шляху від відправника до одержувача. Основна проблема, з якою стикаються користувачі мереж, де застосовується наскрізне шифрування, пов'язана з тим, що службова інформація. використовувана для маршрутизації повідомлень, передається по мережі в незашифрованому вигляді. Досвідчений криптоаналітика може отримати для себе масу корисної інформації, знаючи хто з ким, як довго і в які години спілкується через комп'ютерну мережу. Для цього йому навіть не потрібно бути в курсі предмета спілкування. У порівнянні з канальним, наскрізне шифрування характеризується більш складною роботою з ключами, оскільки кожна пара користувачів комп'ютерної мережі повинна бути забезпечена однаковими ключами, перш ніж вони зможуть зв'язатися один з одним. А оскільки криптографічний алгоритм реалізується на верхніх рівнях моделі OSI, доводиться також стикатися з багатьма суттєвими відмінностями в комунікаційних протоколах і інтерфейсах в залежності від типів мереж і об'єднуються в мережу комп'ютерів. Все це ускладнює практичне застосування наскрізного шифрування.

Комбінований шифрування

Комбінація канального і наскрізного шифрування даних в комп'ютерній мережі обходиться значно дорожче, ніж кожна з них окремо. Однак саме такий підхід дозволяє найкращим чином захистити дані, передані по мережі. Шифрування в кожному каналі зв'язку не дозволяє супротивникові аналізувати службову інформацію, використовувану для маршрутизації. А наскрізне шифрування зменшує ймовірність доступу до незашифрованому даними у вузлах мережі. При комбінованому шифруванні робота з ключами ведеться так: мережеві адміністратори відповідають за ключі, що використовуються при канальному шифруванні, а про ключі, що застосовуються при наскрізному шифруванні, дбають самі користувачі.

8.4.Шифрування файлів

На перший погляд, шифрування файлів можна повністю уподібнити шифруванню повідомлень, відправником та одержувачем яких є одна і та ж особа, а середовищем передачі служить одне з комп'ютерних пристроїв зберігання даних (магнітний або оптичний диск, магнітна стрічка, оперативна пам'ять). Проте все не так просто, як здається на перший погляд.

Якщо при передачі по комунікаційних каналах повідомлення загубиться по дорозі від відправника до одержувача, його можна спробувати передати знову. При шифруванні даних, призначених для зберігання у вигляді комп'ютерних файлів, справи йдуть інакше. Якщо ви не в змозі розшифрувати свій файл, вам навряд чи вдасться зробити це і з другої, і з третьої, і навіть з сотих спроби. Ваші дані будуть втрачені раз і назавжди. Це означає, що при шифруванні файлів необхідно передбачити спеціальні механізми запобігання виникнення помилок в шифртекста.

Криптографія допомагає перетворити великі секрети в маленькі. Замість того щоб безуспішно намагатися запам'ятати вміст величезного файлу, людині достатньо його зашифрувати і зберегти в пам'яті використаний для цієї мети ключ. Якщо ключ застосовується для шифрування повідомлення, то його потрібно мати під рукою лише до тих пір, поки повідомлення не дійде до свого адресата і не буде їм успішно розшифровано. На відміну від повідомлень, шифровані файли можуть зберігатися роками, і протягом усього цього часу необхідно пам'ятати і тримати в секреті відповідний ключ. Є й інші особливості шифрування файлів, про які необхідно пам'ятати незалежно від застосовуваного криптографічного алгоритму: нерідко після шифрування файлу його незашифрованому копія залишається на іншому магнітному диску, на іншому комп'ютері або у вигляді роздруківки, зробленої на принтері; розмір блоку у блочному алгоритмі шифрування може значно перевищувати розмір окремої порції даних в структурованому файлі, в результаті чого зашифрований файл виявиться набагато довше початкового; швидкість шифрування файлів за допомогою обраного для цієї мети криптографічного алгоритму повинна відповідати швидкостями, на яких працюють пристрої введення / виведення сучасних комп'ютерів; робота з ключами є досить непростою справою, оскільки різні користувачі повинні мати доступ не тільки до різних файлів, але і до окремих частин одного і того ж файлу. Якщо файл являє собою єдине ціле (наприклад, містить відрізок тексту), відновлення цього файлу в початковому вигляді не зажадає великих зусиль: перед використанням достатньо буде просто розшифрувати весь файл. Однак якщо файл структурований (наприклад, розділений на записи і поля, як це робиться в базах даних), то розшифрування всього файлу цілком кожен раз, коли необхідно здійснити доступ до окремої порції даних, зробить роботу з таким файлом надзвичайно неефективною. Шифрування порцій даних в структурованому файлі робить його вразливим по відношенню до атаки, при якій зловмисник відшукує в цьому файлі потрібну порцію даних і замінює її на іншу за своїм розсудом.

У користувача, який хоче зашифрувати кожен файл, размешение на жорсткому диску комп'ютера, є дві можливості. Якщо він використовує один і той же ключ для шифрування всіх файлів, то згодом опинимося не в змозі розмежувати доступ до них з боку інших користувачів. Крім того, в результаті у криптоаналітика буде багато шифртекста, отриманого на одному ключі, що істотно полегшить розтин цього ключа. Краще шифрувати кожен файл на окремому ключі, а потім зашифрований, всі ключі за допомогою майстер-ключа. Тим самим користувачі будуть врятовані від суєти, пов'язаної з організацією надійного зберігання безлічі ключів. Розмежування доступу груп користувачів до різних файлів буде здійснюватися шляхом розподілу безлічі всіх ключів на підмножини і шифрування цих підмножин на різноманітних майстер-ключах. Стійкість такої криптосистеми буде значно вище, ніж у випадку використання єдиного ключа для шифрування всіх файлів на жорсткому диску, оскільки ключі, застосовувані для шифрування файлів, можна генерувати випадковим чином і, отже більш стійкими проти словникової атаки.

8.5.Апаратне і програмне шифрування

Апаратне шифрування.

Більшість засобів криптографічного захисту даних реалізовано у вигляді спеціалізованих фізичних пристроїв. Ці пристрої вбудовуються в лінію зв'язку і здійснюють шифрування всієї переданої але ній інформації. Переважання апаратного шифрування над програмним обумовлено декількома причинами. Більш висока швидкість. Криптографічні алгоритми складаються з величезного числа складних операцій, які виконуються над бітами відкритого тексту. Сучасні універсальні комп'ютери погано пристосовані для ефективного виконання цих операцій. Спеціалізоване обладнання вміє робити їх набагато швидше. Апаратуру легше фізично захистити від проникнення ззовні. Програма. виконувана на персональному комп'ютері, практично беззахисна. Озброївшись відладчиком, зловмисник може внести в неї приховані зміни, щоб знизити стійкість використовуваного криптографічного алгоритму, і ніхто нічого не помітить. Що ж стосується апаратури, то вона зазвичай поміщається в особливі контейнери, які унеможливлюють зміну схеми її функціонування. Чіп покривається спеціальним хімічним складом, і в результаті будь-яка спроба подолати захисний шар цього чіпа призводить до самознищення його внутрішньої логічної структури. І хоча іноді

електромагнітне випромінювання може служити хорошим джерелом інформації про те, що відбувається всередині мікросхеми, від цього випромінювання легко позбутися, заекранірован мікросхему. Аналогічним чином можна заекранірован і комп'ютер, проте зробити це набагато складніше, ніж мініатюрну мікросхему. Апаратура шифрування більш проста в установці. Дуже часто шифрування потрібно там, де додаткове комп'ютерне обладнання є абсолютно зайвим. Телефони, факсимільні апарати та модеми значно дешевше обладнати пристроями апаратного шифрування, ніж вбудовувати в них мікрокомп'ютери з відповідним програмним забезпеченням. Навіть в комп'ютерах установка спеціалізованого шифрувального обладнання створює менше проблем, ніж модернізація системного програмного забезпечення з метою додати до нього функцій шифрування даних. В ідеалі шифрування повинно здійснюватися непомітно для користувача. Щоб добитися цього за допомогою програмних засобів, засоби шифрування повинні бути сховані глибоко в надра операційної системи. З готової і налагодженої операційною системою виконати це безболісно не так-то просто. Але навіть будь непрофесіонал зможе приєднати шифрувальний блок до персонального комп'ютера, з одного боку, і до зовнішнього модему, з іншого. Сучасний ринок апаратних засобів шифрування інформації пропонує потенційним покупцям 3 різновиди таких засобів - самодостатні шифрувальні модулі (вони самостійно виконують всю роботу з ключами), блоки шифрування в каналах зв'язку і шифрувальні плати розширення для установки в персональні комп'ютери. Більшість пристроїв першого і другого типів є вузько спеціалізованими. і тому перш, ніж приймати остаточне рішення про їх придбання, необхідно досконально вивчити обмеження, які при установці накладають ці пристрої на відповідне "залізо", операційні системи та прикладне програмне забезпечення. А інакше можна викинути гроші на вітер, ні на йоту не наблизившись до бажаної мети. Правда, іноді вибір полегшується тим, що деякі компанії торгують комунікаційним устаткуванням, яке вже має встановлену апаратуру шифрування даних. Плати розширення для персональних комп'ютерів є більш універсальним засобом апаратного шифрування і зазвичай можуть бути легко сконфігуровані таким чином, щоб шифрувати всю інформацію, яка записується на жорсткий диск комп'ютера, а також усі дані, що пересилаються на дискети і в послідовні порти. Як правило, зашита від електромагнітного випромінювання в шифрувальних платах розширення відсутня, оскільки немає сенсу захищати ці плати, якщо аналогічні заходи не робляться щодо всього комп'ютера.

Програмне шифрування

Будь криптографічний алгоритм може бути реалізований у вигляді відповідної програми. Переваги такої реалізації очевидні: програмні засоби шифрування легко копіюються, вони прості у використанні, їх неважко модифікувати відповідно до конкретних потреб. У всіх поширених

операційних системах є вбудовані засоби шифрування файлів. Зазвичай вони призначені для шифрування окремих файлів, і робота з ключами цілком покладається на користувача. Тому застосування цих засобів потребує особливої уваги. По-перше, в жодному разі не можна зберігати ключі на диску разом з зашифрованими з їх допомогою файлами, а по-друге, незашифровані копії файлів необхідно видалити відразу після шифрування. Звичайно, зловмисник може дістатися до комп'ютера і непомітно внести небажані зміни в програму шифрування. Однак основна проблема полягає зовсім не в цьому. Якщо зловмисник в змозі проникнути в приміщення, де встановлено комп'ютер, він навряд чи буде возитися з програмою, а просто встановить приховану камеру в стіні, підслуховуючий пристрій - в телефон або датчик для ретрансляції електромагнітного випромінювання - в комп'ютер. Зрештою, якщо зловмисник може безперешкодно все це зробити, бій з ним програно, навіть ще не розпочавшись.

8.6.Стискання та шифрування

Алгоритми стиснення даних дуже добре підходять для спільного використання з криптографічними алгоритмами. Тому є дві причини: При розтині шифру криптоаналітика, як правило, покладається на надмірність, властиву будь-якого відкритого тексту. Стиснення допомагає позбутися від цієї надмірності. Шифрування даних є досить трудомісткою операцією. При стисненні зменшується довжина відкритого тексту, за рахунок чого скорочується час, який буде витрачено на його шифрування. Треба тільки не забути стиснути файл до того, як він буде зашифрований. Після шифрування файлу за допомогою якісного криптографічного алгоритму отриманий шифртекст стиснути не вдасться, оскільки його характеристики будуть близькі до характеристик абсолютно випадкового набору букв. До речі, стиснення може служити своєрідним тестом для перевірки якості криптографічного алгоритму: якщо шифртекст піддається стисненню, значить цей алгоритм краще замінити на більш досконалий.

8.7.Як заховати один шифртекст в іншому

Антон і Борис кілька місяців обмінювалися шифрованими повідомленнями. Контррозвідка перехопила всі ці повідомлення, але так і не змогла прочитати ні єдиного слова. Контррозвідникам набридло колекціонувати листування Антона і Бориса, не знаючи її змісту, і вони вирішили заарештувати підозрілу парочку. Перший же допит почався словами: "Де ключі до шифру?" "До якого такого шифру?!" - В один голос вигукнули Антон і Борис, але тут же осіклася і зблідли, помітивши на столі у слідчого зловісного виду кліщі, вкриті плямами чи іржі, чи то крові. Антон і Борис змогли б викрутитися із ситуації, якщо б шифрували кожне своє повідомлення так, щоб воно

допускало два різних расшифрования в залежності від використовуваного ключа. Своє справжнє секретне повідомлення Борису Антон міг би зашифрувати на одному ключі, а цілком невинний відкритий текст - на іншому. Тепер, якщо від Антона зажадають ключ до шифру, він віддасть підставний ключ, який дозволить прочитати безневинне повідомлення, а інший ключ збереже в таємниці. Найпростіший спосіб зробити це потребує використання одноразового блокнота. Нехай Р - секретний відкритий текст, D - невинний відкритий текст, С - шифрований текст, К - справжній ключ, а К - підставний ключ. Антон шифрує Р:Р Å К = С Оскільки у Бориса є копія ключа К, він може без проблем розшифрувати повідомлення Антона: З Å К = Р Якщо контррозвідники спробують змусити Антона і Бориса видати вони використовують ключ, то замість До вони можуть повідомити в контррозвідку: К '= С Å D У результаті контррозвідники зможуть прочитати безневинний відкритий текст: З Å К '= D Так як Антон і Борис користуються одноразовим блокнотом, то До є повністю випадковим і довести, що К є підставною ключем, практично неможливо (не вдаючись до тортур). Антон міг би зашифрувати Р не за допомогою одноразового блокнота, а користуючись будь-яким із своїх найулюбленіших криптографічних алгоритмів і ключем К. Склавши С з фрагментом будь-якого загальновідомого твору (наприклад, з уривком з другого розділу "Ідіота") по модулю 2, Антон отримає К '. Тепер якщо до Антона пристануть злі "дядечки" з контррозвідки, він пред'явить їм С разом з К 'і скаже, що К' - це одноразовий блокнот для С і що він просто захотів попрактикуватися н криптографії, зашифрувавши для цієї мети уривок з першою-ліпшою книги . І поки контррозвідники не отримають в своє розпорядження ключ К, довести, що Антон займався чимось протизаконним, вони не зможуть.

8.8Чому криптосистеми ненадійні

В даний час криптографія успішно використовується майже у всіх інформаційних системах - від Internet до баз даних. Без неї забезпечити необхідну ступінь конфіденційності в сучасному, до межі комп'ютеризованому світі вже не представляється можливим. Крім того, за допомогою криптографії запобігають спроби шахрайства в системах електронної комерції і забезпечується законність фінансових угод. З часом значення криптографії, ймовірно, зросте. Для цього припущення є вагомі підстави. Однак з прикрістю доводиться визнати, що переважна більшість криптографічних систем не забезпечує того високого рівня захисту, про який із захопленням зазвичай говориться в їх рекламі. Багато з них до цих пір не були зламані з тієї простої причини, що поки не знайшли широкого розповсюдження. Як тільки ці системи почнуть повсюдно застосовуватися на практиці, вони, немов магніт, стануть залучати пильну увагу зловмисників, яких сьогодні розвелося безліч. При цьому удача й везіння будуть явно на боці останніх. Адже для досягнення своїх цілей їм достатньо знайти в

захисних механізмах всього лише одну пролом, а обороняється доведеться зміцнювати всі без винятку вразливі місця.

Реалізація

Зрозуміло, що ніхто не в змозі надати стовідсоткову гарантію безпеки. Тим не менш, криптографічний захист без особливих зусиль можна спроектувати так, щоб вона протистояла атакам зловмисників аж до того моменту, коли їм стане простіше добути бажану інформацію іншим шляхом (наприклад, за допомогою підкупу персоналу або впровадження програм-шпигунів). Адже криптографія дійсно хороша саме тим, що для неї вже давно придумані ефективні алгоритми та протоколи, які необхідні, щоб надійно захистити комп'ютери і комп'ютерні мережі від електронного злому і проявів вандалізму. Ось чому в реальному житті криптографічні системи рідко зламуються чисто математичними методами. Адже криптографічний алгоритм або протокол від його практичної реалізації у вигляді працюючої програми, як правило, відокремлює зяюча прірву. Навіть доведений за всіма правилами формальної логіки факт, що криптографічний зашита досконала з математичної точки зору, зовсім не означає, що вона залишиться такою після того, як над її впровадженням попрацюють програмісти. Відомо, що під тиском бюджетних обмежень, дефіциту часу і особистих негараздів програмісти неминуче допускають дуже серйозні помилки при реалізації алгоритмів - використовують погані датчики випадкових чисел для генерації криптографічних ключів, не враховують специфіку апаратної середовища, у якій доведеться експлуатувати створені ними програмні засоби, а також регулярно забувають видаляти ключову та іншу секретну інформацію з оперативної пам'яті комп'ютера або з магнітного носія після того, як потреба в її зберіганні там відпала. Єдиний спосіб навчитися уникати цих та їм подібних помилок полягає в тому. щоб знову і знову намагатися створити досконаліші системи криптографічного захисту даних, а потім не менш завзято намагатися їх зламати. Звичайно, після того як пролом в системі криптографічного захисту знайдена, її досить легко можна залатати. Але сам пошук подібного роду дефектів є неймовірно складним завданням. Ніяке попереднє тестування не допоможе виявити в криптографічного системі всі дефекти, оскільки жоден тест окремо не може дати повної гарантії їх відсутності. Адже якщо програма шифрування правильно зашифровує і розшифровує файли, це ще зовсім не означає, що вона надійно захищає їх вміст.

Облік реальних потреб користувачів

Чимало проблем, пов'язаних з використанням криптографічних засобів, створюють самі користувачі. Безпека турбує їх менше за все. У першу чергу їм потрібні простота, зручність і сумісність з вже існуючими (як правило, недостатньо захищеними) програмними продуктами. Вони вибирають легко запам'ятовуються криптографічні ключі, записують їх де попало, запросто

діляться ними з друзями та знайомими. Тому грамотно спроектована криптографічна система обов'язково повинна брати до уваги специфічні особливості поведінки людей. Ще важче виявляється переконати людей в необхідності строго і неухильно застосовувати криптографічний захист даних. Користувачі охоче приносять у жертву власну безпеку, якщо засоби її забезпечення заважають їм скоріше зробити свою роботу. Тому тільки у випадку, якщо при проектуванні криптографічного системи були враховані реальні потреби користувачів, вона дійсно в змозі захистити їхні комп'ютери і комп'ютерні мережі.Законодавчі обмеження У Зводі законів США є пункт 2778, який називається "Контроль за експортом та імпортом озброєнь". Саме цей пункт є юридичною основою для низки інструкцій, що іменуються "Правилами контролю за переміщенням зброї в світі" (International Traffic in Arms Regulations, скорочено-1TAR). Розділ 120.1 ITAR прямо зараховує до військового спорядження, за переміщенням якого Сполучені Штати осу шести, (я ють найсуворіший контроль, програмне забезпечення, призначене для цілей ефективного шифрування даних. А це означає, що американським компаніям, що бажають експортувати програми ефективної) шифрування, необхідно зареєструватися у Державному департаменті США в якості торговця військовим майном та отримати там ліцензію на експорт. Відомо, що при видачі таких ліцензій Держдепартамент цілком і повністю покладається на думку АНБ. У результаті ліцензія на експорт криптографічних засобів нікому не видається до тих пір, поки АНБ не схвалить таке рішення. У свою чергу, АНБ аж ніяк не зацікавлене у вільному поширенні надійних програм шифрування за межами країни. Тому всі програмні засоби, вироблені в США і легально експортуються за кордон, забезпечують ослаблену криптографічний захист. Щоб підвищити свою конкурентоспроможність на світовому ринку, виробники засобів криптографічного захисту у США змушені шукати лазівки в законодавстві. Наприклад, відома американська фірма RSA Data Security спробувала обійти закон шляхом фінансування зусиллі китайських вчених, яких уряд Китаю офіційно уповноважив розробити нові програмні засоби шифрування даних. Передбачалося, що ці кошти, створені на основі алгоритмів, переданих американською фірмою китайцям, зможуть забезпечити більш надійну криптографічний захист інформації, ніж ті, які Китай в стані імпортувати з США відповідно до чинного американського законодавства. Це, безсумнівно, радісна подія для Китаю, проте слід зазначити, що заради задоволення потреб пересічного користувача за кордоном, що не володіє можливостями і ресурсами, порівнянними з тими, які є в розпорядженні китайського уряду, американські виробники програм ефективного шифрування навряд чи будуть шукати якісь або шляхи, що ведуть в обхід американського законодавства. Наслідуючи приклад США, ряд держав, ввели обмеження на експорт, імпорт і використання шифрувальних засобів. Тим не менш, багатьох українських громадян нітрохи не лякають законодавчі обмеження на експлуатацію шифрувальних засобів. Вони твердо дотримуються думки про те, що

належить ним інформація безумовно є об'єктом їхньої власності, і що вони, як власники своєї інформації, мають право самостійно визначати правила її зберігання і захисту. Залишається тільки зі знанням справи вирішити, які саме шифрувальні засоби застосовувати для адекватного захисту цієї інформації, а які не використовувати ні в якому разі, зважаючи на їх слабкою надійності.

Занадто мала довжина ключа

Занадто мала довжина ключа - одна з найбільш очевидних причин ненадійності криптографічних систем. Причому недостатню довжину ключа можуть мати навіть ті криптосистеми, в яких застосовуються самі надійні алгоритми шифрування, оскільки: в них спочатку може бути присутнім можливість роботи з ключем змінної довжини для того, щоб при використанні цих систем на практиці можна було вибрати потрібну довжину ключа, виходячи з бажаної надійності та ефективності; вони розроблялися тоді, коли дана довжина використовуваного ключа вважалася більш ніж достатньою для забезпечення необхідного рівня криптографічного захисту; на них поширюються експортні обмеження, які встановлюють допустиму довжину ключа на рівні, що не відповідає сучасним вимогам. Першим надійним криптографічним алгоритмом, який впритул зіштовхнувся з проблемою вибору адекватної довжини ключа, став RSA. Справа в тому, що його розтин вимагає розкладання на множники (факторизації) дуже великих чисел. У березні 1994 р. за цілком прийнятний час було факторізовано 428-бітове число, а на сьогоднішній день досить реальним представляється факторизація 512-бітових чисел. Досягнутий прогрес у вирішенні завдання факторизації дуже великих чисел пов'язаний не тільки із зростанням обчислювальних потужностей сучасного комп'ютерного парку, але і з розробкою нових ефективних алгоритмів. На тому, що це завдання є дуже трудомісткою, ще зовсім недавно була заснована надійність криптографічного алгоритму, що використовується в поширеній програмі PGP. Тому можна стверджувати, що сьогодні розкладання на множники є однією з найбільш динамічно розвиваються областей криптографії. На початку 1998 р. через занадто малу довжину ключа (56 біт) фактично "наказав довго жити" DES-алгоритм, який тривалий час був офіційним стандартом шифрування даних у США. Зараз американським Національним інститутом стандартів оголошений конкурс на новий стандарт шифрування даних Advanced Encryption Standard (AES). Згідно з умовами цього конкурсу, кандидати на роль AES повинні представляти собою симетричні алгоритми шифрування з ключем довжиною понад 128 біт.

Потаємні ходи

Причини появи потайних ходів в криптографічних системах досить очевидні: їх розробники хочуть мати контроль над шіфруемий в цих системах

інформацією і залишають для себе можливість розшифровувати її, не знаючи ключа користувача. Засіб, за допомогою яких дана можливість реалізується на практиці, і прийнято іменувати потайним ходом. Іноді потаємні ходи застосовуються для цілей налагодження, а після її завершення розробники в поспіху просто забувають прибрати їх з кінцевого продукту. Класичний приклад потайного ходу, який хакерами одностайно визнається найталановитішим "хаком" по злому системи парольного захисту всіх часів і народів, привів Кен Томпсон (один з авторів компілятора для мови програмування С) у своїй лекції з нагоди вручення йому престижної премії Тьюринга. Справа в тому, що в операційній системі UNIX користувача паролі зберігаються в зашифрованому вигляді у спеціальній базі даних. У компілятор мови С Томпсоном був завбачливо вставлений код, розпізнавати, коли на вхід компілятора надходила програма, яка містила запрошення користувачеві зареєструватись (login). Тоді компілятор додавав до цієї програми код, який розпізнавав пароль, обраний самим Томпсоном. Таким чином, Томпсон отримував можливість успішно проходити процедуру реєстрації та ідентифікації, не знаючи легальних паролів, що зберігаються в зашифрованою базі даних. Стандартний спосіб закрити такий потаємний хід полягає в тому, щоб видалити з вихідного тексту компілятора "шкідливий" код, а потім його перекомпілювати. Але при перекомпіляції знову не обійтися без компілятора. І Томпсон дописав свій компілятор так, щоб той розпізнавав, коли на його вхід надходила виправлена версія його самого. У цьому випадку компілятор додавав у неї код, який, у свою чергу, при компіляції програм із запрошенням login дописував в них код, який дає Томпсону привілейований доступ, а також код, який дозволяв компілятору розпізнавати свою оновлену версію при перекомпіляції. Таким чином, не має значення, наскільки надійним був криптографічний алгоритм, який використовувався для шифрування паролів користувачів операційної системи UNIX. Потаємний хід, придуманий Томпсоном. залишався відкритим для нього за будь-яких умовах.

Шифрування навколо нас

Отже, для того щоб створити надійну криптографічну систему, необхідно володіти достатніми знаннями в галузі сучасної криптографії, акуратно і безпомилково втілити ці знання у вигляді працюючої програми з дружнім інтерфейсом, прибравши з неї всі потаємні ходи після закінчення налагодження. Інші криптосистеми, пропоновані закордонними фірмами у вигляді закінчених продуктів або бібліотек, включаючи як встояли закордонні стандарти, так і самостійні оригінальні розробки, є незаконними. Якщо припустити, що шифрування - це така нестандартна кодування даних, яка серйозно ускладнює можливість їх перекодування в стандартне уявлення без відповідного апаратного або програмного забезпечення, то в категорію шифрсистем тут же потраплять архіватори (pkzip, arj і r a r), відомі текстові редактори (Word і Lexicon), а також засоби редагування графічних

зображень (Paint і CorelDraw), оскільки всі вони використовують свою власну нестандартну кодування, що не дозволяє без відповідних програм переглядати закодовані з їх допомогою дані. Спроба придумати універсальний критерій підрозділи кодувань на стандартні і нестандартні заздалегідь приречена на провал, оскільки розробників програмного забезпечення не можна змусити користуватися тільки кодуванням, схваленої указом президента, як стандартної. Тому краще до систем шифрування відносити, наприклад, програмні засоби, до яких додається документація з явним зазначенням того факту, що вони призначені саме для шифрування даних. Враховуючи плутанину, що панує в російському законодавстві, не дивно, що російські користувачі для захисту своєї конфіденційно інформації активно застосовують архіватори з парольного захистом, Norton Diskreet, Word, Excel, численні умовно-безкоштовні програми (PGP, CodeDvag, SecurPC, Secur-all 32, BestCrypt NP, Kremlin та ін.) криптографічні системи вітчизняних фірм , власні кустарні розробки, а також програми невідомого походження. Більшість з них вкрай слабкі, і програми їх злому за цілком помірну плату можна отримати, наприклад, в Internet. Виняток у списку свідомо ненадійних криптографічних систем, складають лише кілька оригінальних розробок фірм. Проте зважаючи спорудженої нашою державою інформаційної блокади навколо криптографії і всього, що з нею пов'язано, можна тільки робити припущення, які саме. Підводячи підсумок сказаному, можна зробити висновок про те, що ситуація па ринку криптографічних систем не вселяє оптимізму. Законодавчі обмеження, помилки в реалізації, недружній інтерфейс, недостатня довжина ключа та наявність потайних ходів призводять до того, що відшукати надійну криптосистему практично неможливо. Оскільки криптографія покликана обслуговувати потреби людства в досить делікатній сфері (за допомогою криптографічних методів зберігається в таємниці конфіденційна інформація з обмеженим, на думку її власників, безконтрольного розповсюдження), деякі дослідники вбачають у ситуації, що склалася дію певних таємних сил, які намагаються направляти та контролювати прогрес людства в області криптографії. Одна з головних турбот цих таємних сил - взяти кожного "під ковпак", тобто мати найбільш повне досьє на будь-яку людину. Тому таємні суспільства так зацікавлені в одноосібному володінні елітарними криптографічними знаннями та створеними на основі цих знань надійними засобами криптографічного захисту даних, безконтрольне поширення яких може поставити під загрозу їх здатність ведення тотального стеження. Безрезультатно закінчуються багатообіцяючі криптографічні дослідження, при загадкових обставинах обриваються життя талановитих криптографов, виникають всілякі перепони на шляху вільного обміну інформацією про останні криптографічних пошуках. Інші дослідники закулісних пружин історії йдуть ще далі і стверджують, що саме найбільш повні і достовірні знання з області криптології (науки, що об'єднує криптографію і криптоаналіз), дозволили нинішнім таємним володарям, який розпоряджається долею людства, досягти

вершин своєї могутності. Гіпотеза цих дослідників полягає в тому, що криптология є одним з ефективних інструментів пізнання навколишнього світу: інформація про головні напрямки його розвитку в зашифрованому вигляді доступна кожному і її можна витягти шляхом дешифрування. Хто знає, як це робиться, має майже необмежену владу над світом, оскільки може з великою вірогідністю передбачати майбутнє. Об'єднані чи таємні верховні правителі в єдину організацію? Швидше за все, між їх різними спільнотами існує серйозна конкуренція. Та й могутність їх простягається до певних меж. Тому час від часу цілком вірогідна поява надійних криптосистем, хоча б на обмежений період часу. Перевірити ці гіпотези на практиці видається неможливим, і деяким вони можуть здатися занадто сміливими, але мати про них уявлення абсолютно необхідно. Хоча б для того, щоб у разі придбання вами свідомо ненадійною криптосистеми для потреб вашої фірми або організації виправдати свою помилку перед керівництвом втручанням якихось таємних всемогутніх сил.

Література

1.Ахраменко М.Ф. Проблеми криміналізації суспільно-небезпечної поведінки з використанням інформаційно-обчислювальних систем: Автореф. дис. ... к.ю.н.: 12.00.08/ БДУ. - Мінськ, 1996. - С.7.

2. Галицкий А.В., Рябко С.Д., Шаньгин В.Ф. Защита информации в сети. – М.:ДМК Пресс, 2004.

3. Голубєв В.О. Удосконалення боротьби зі злочинами у сфері використання автоматизованих електронно-обчислювальних систем // Боротьба з організованою злочинністю і корупцією (теорія і практика). Науково-практичний журнал. - № 3. - Київ, 2001. - С.171-172.

3. . Голубев В.А. Подписание конвенции “По борьбе с киберпреступностью” и некоторые проблемы расследования киберпреступлений. - http://www/crime-research.org/library/convention.htm.

4.Жельников В. «Криптография от папируса до компьютера» - М., 1996

5.Законодавство України про інформацію. К., 1998.

6.Казаков С.И. Основы сетевых технологий. М.: Академич. экспрес., 1998, с.456.

7. Кашинская А.Н. Зарубежный опыт правового регулирования использования Internet // Управление защитой информации. - 2000. - Т.4. - № 2. - С.71-76.

8. Компьютерная преступность – угроза национальной безопасности // http://www.crime-research.org/news/2003/07/1402.html. 14.07.03.

9. Крилов В.В. Інформаційні комп’ютерні злочини. - М.: ИНФРА-М-НОРМА, 1997. - С.11.

10. Копылов В.А. Информационное право. М., 1997.

11.Крюкова Е.П. Правові аспекти забезпечення інформаційної безпеки // Комплексний захист інформації: Тези доповіді на IV Міжнар. конф., Мінськ, 29.02.-02.03.2000 р. - М.: Ось-89, 1996. - С.214.

12. М.А. Деднев, Д.В. Дыльнов, М.А. Иванов Защита информации в банковском деле и электронном бизнесе. М.: Кудиц-образ, 2004. – 512 с.

13. Мафтик С. Механизм защиты в сетях ЭВМ. М.: Мир, 1993.

14.Проскурин В.Г., Крутов С.В., Мацкевич И.В. Защита в операционных системах. – М.: «Радио и связь», 2000.

15. Попередження комп’ютерних злочинів // Проблеми злочинності в капіталістичних країнах (За матеріалами зарубіжної преси). - М.: ВНДІ МВС СРСР, 1986, № 4. - С.10.

16. Попередження комп’ютерних злочинів // Проблеми злочинності в капіталістичних країнах (За матеріалами зарубіжної преси). - М.: ВНДІ МВС СРСР, 1986, № 4. - С.4.

17.Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. – СПб.: Наука и техника, 2004.

18.Щербаков А, Домашев А. Прикладная криптография. Использование и синтез криптографических интерфейсов. М.:Русская редакция, 2003.

19. Ярочкін В.І. Безпека інформаційних систем. - М.: Ось-89, 1996. - С.214.

20.http://www.des-crypto.ru/itsecur/password/.
21.http://www.hardline.ru/selfteachers/Info/Security/Protection_to_information/1/Index.htm

Іван Федорович Чернецький
спеціаліст системотехнік, магістрант
інформаційних технологій

ТЕХНОЛОГІЇ КОМП'ЮТЕРНОЇ БЕЗПЕКИ
КНИГА 5
ІН 11М

Комп'ютерний набір, верстка і макетування та
дизайн в редакторі Microsoft®Office® Word 2003
І.Ф.Чернецький. Науковий керівник Р. М. Літнарович,
доцент, кандидат технічних наук
Міжнародний Економіко-Гуманітарний Університет ім.
акад. Степана Дем'янчука
Кафедра математичного моделювання
33027,м.Рівне,Україна
Вул.акад. С.Дем'янчука,4, корпус 1
Телефон:(+00380) 362 23-73-09
Факс:(+00380) 362 23-01-86
E-mail:mail@regi.rovno.ua
E-mail: chernetskiy_2011@mail.ru