СИГУРНА ЕЛЕКТРОННА ПОЩА

Мрежова сигурност

Изготвили:

Николай Любенов Филчев
информатика 4-ти курс
Фак.No.: 42370

Станислав Маринов Бенов
математика и информатика 3-ти курс
Фак.No.: 13064

1.Обзор на потока на e-mail

Електронната поща (e-mail) е мрежова услуга, чрез която всеки потребител, свързан към глобална компютърна мрежа, може да обменя съобщения с всички потребители на мрежата. Тази услугата придоби огромна популярност поради факта, че осигурява надеждна и относително бърза връзка между потребители в целия свят. Чрез нея могат да бъдат изпращани или получавани данни в различен формат. Обикновено обменяната информация е в текстов формат.

Съвременните програми, използувани за изпращане и получаване на електронни писма, позволяват да се обменят и друг вид данни посредством тяхното кодиране по определени стандарти. Съществуват също така възможности за компресиране на предаваните файлове, което улеснява изпращането и получаването на големи по обем масиви от данни.

Основните понятия, свързани с електронната поща, са следните:

• **пощенски сървър** *(mail server)* - представлява програмна система, реализираща основните функции на електронната поща. За всеки домейн от глобалната мрежа е необходим поне един такъв сървър. Тази система включва следните основни компоненти:

• **пощенска кутия** *(mail box)* - специализиран файл, съхраняван в определена директория на сървъра. Всеки такъв файл носи името на съответния потребител и е предназначен да съхранява пристигащата за този потребител електронна поща.

• **прехвърлящ хост** *(relay host)* - програма, която управлява маршрутизирането и изпращането на електронните писма в рамките на глобалната мрежа. Тя обработва в последователен ред писмата, съхранявани в опашката за изходящи писма на сървъра. При изпращане на маршрутизираното писмо се използува пощенският транспортен агент.

• **пощенски транспортен агент** *(mail transport agent -MTA)* - програма, която приема и изпраща електронните писма. При приемане на дадено писмо тя анализира адреса на получателя. Ако получателят има пощенска кутия на сървъра, писмото се записва в нея. В противен случай то се записва в опашката за изходящи писма.

• **пощенски клиент** *(mail client)* - програмна система, която изпраща/получава електронни писма към/от пощенския сървер.

• **шлюз** *(gateway)* - програмна система, която управлява обмена на електронна поща между различни видове комуникационни мрежи.

Поради широкото разпространение на Интернет в момента такива системи почти не се използуват.

Основни принципи на действие

Създаването на писмо, изпращането му чрез електронна поща и прочитането на пристигналите в пощенската кутия на потребителя електронни писма се осъществява обикновено посредством *Пощенски Клиент - ПК (mail client)*. ПК е разпространеното наименование на програма, изпълняваща тези функции. В зависимост от използувания компютър и съответната операционна система съществува голямо разнообразие от такива програми, като например Mail Tool и mailx за UNIX операционна система или MS Outlook Express, Peqassus Mail, Eudora за IBM персонални компютри.

На картинката са показани основните етапи, през които преминава изпращането на едно електронно писмо, в случая, когато изпращачът и получателят имат пощенски кутии на различни сървъри.

Посредством ПК потребителят подготвя и изпраща електронното писмо. При изпращането ПК се свързва по протокола SMTP с ПТА на пощенския сървър на потребителя. ПТА приема писмото и тъй като то е предназначено за получател с пощенска кутия на друг сървър го записва в опашката за изходящи писма. ПХ обработва опашката и при достигане на това писмо го маршрутизира. За целта той използува DNS при определяне на пощенския сървър на получателя. След това той

предава писмото на своя ПТА, който се свързва по протокола SMTP с ПТА на пощенския сървър на получателя. Този ПТА приема писмото и го записва в пощенската кутия на получателя. Получателят може да прочете това писмо в произволен момент на времето. За целта той трябва да използува собствения си ПК, който прехвърля писмото от кутията на пощенския сървър в компютъра на получателя, използувайки протокола POP3.

Основни функции на протоколите и програмите, реализиращи електронна поща

За обмен на данни чрез електронна поща се използуват множество протоколи по-важните от които са:

• **SMTP** (Simple Mail Transfer Protocol) - този протокол осигурява обмен на писмата между програмите, предназначени за изпращане и получаване на електронна поща.

• **POP** (Post Office Protocol) - този протокол е предназначен за прехвърляне съдържанието на пощенската кутия на потребителя от пощенския сървър към персоналния компютър на потребителя. Актуалната версия на този протокол е 3.

• **IMAP** (Internet Message Access protocol) - протоколът осигурява връзка между пощенския сървър и потребителски работни станции или персонални компютри чрез динамичен достъп до пощенските кутии на сървъра Разликата с протокола POP е, че прочетените писма остават на съхранение в пощенския сървър, а не се прехвърлят в локалната система. Това позволява на потребителя да има достъп до пощенската си кутия от различни клиентски компютри. Освен това в пощенската кутия на сървъра се поддържа йерархична система от директории, част от които могат да бъдат обявени и като публични. Възможно е търсене на писмо по някакъв признак и преглеждане само на заглавните части на писмата без да бъдат изтегляни изцяло на клиентския компютър. Актуалната версия на този протокол е 4.

• **UUCP** (Unix to Unix CoPy) - опростен протокол за обмен на електронни съобщения между компютърни системи, работещи с UNIX операционна система.

• **X. 400** - протокол е приет от ITU (International Telecommunicaton Union) и ISO (International Standartisation Organisation) и стандартизиран като ISO 10021. В днешно време е по-малко разпространен, за сметка на SMTP.

Форматът на пощенските съобщения, обменяни в Интернет, е дефиниран в RFC 822, а протоколът UUCP - в RFC 976.

Повечето съвременни операционни системи използват програмата sendmail като пощенски сървер. Тя е предназначена да обслужва получаването и доставката на електронните писма. Програмата Sendmail приема писмата от различни пощенски клиенти, презаписва адреса на получателя в подходяща за анализ форма, анализира го, маршрутизира и изпраща пощата до пощенския сървър на получателя. Освен това тя получава писмата, предназначени за локалните потребители и ги разпределя в техните пощенски кутии, като дава възможност за използуването на псевдоними. При необходимост от изпращане на едно и също съобщение до голям брой потребители, програмата sendmail позволява използуването на пощенски списъци (mailing lists)

В последните години се създадоха множество потребителски програми, осигуряващи графичен интерфейс на потребителя и улесняващи изпращането и приемането, както на прости текстови съобщения, така също и на присъединени съобщения (attachment). Присъединеното съобщение може да бъде обикновен всякакъв файл.
За да могат да се обменят двоични файлове чрез електронна поща, е необходимо те да се кодират по подходящ начин. За кодиране на електронни съобщения се използува един от следните стандарти :

• **MIME** (Multipurpose Internet Mail Extensions) стандарт, описващ формата на данните в тялото на електронното съобщение.

• **Base 644** представлява схема за кодиране на символите в електронното съобщение съгласно изискванията на MIME стандарта. С помощта на 64 символа, които са подмножество от символите на International Alphabet(IA5), е постигнат метод за универсално кодиране, който гарантира независимост по отношение на локално използуваната кодова таблица ASCII или EBCDIC.

• **Uuencode** е метод на кодиране на електронните съобщения, предназначен за трансфер на двоични данни в компютърни системи, поддържащи само ASCII кодовата таблица, при която един символ се кодира със седем бита. При прехвърлянето на данни, които са кодирани чрез този метод, през пощенски шлюзове, осъществяващи прекодиране от ASCII към EBCDIC кодова таблица, е възможно да настъпи нарушаване на структурата на прехвърляните данни.

За крайния потребител на услугата електронна поща всички подробности за начините на представяне на данните, тяхното кодиране, компресиране и трансфер остават скрити, тъй като се изпълняват автоматично след кратка предварителна настройка на

пощенските клиенти. Основните възможности на повечето съвременни програми пощенски клиенти са :

• изпращане/получаване на електронни съобщения.

• възможност за изпращане/получаване на присъединени съобщения (attachment),

• възможност за изпращане на писма до повече от двама потребители едновременно чрез ползуване на пощенски списъци (mailing lists).

• създаване на списък на най-често използуваните пощенски адреси (bookmark address) и избор на псевдоним за всеки от тях, като псевдонимът може след това да служи за заместител на дълги и трудни за запомняне пощенски адреси.

• възможност за ползуване на отдалечен пощенски сървър чрез POP3 или IMAP4 потребителско име.

• възможност за едновременното обслужване на пощенски потребителски имена, разположени на няколко различни пощенски сървъра (multiuse mail support).

Поради широкото използуване на услугата електронна поща голяма част от функциите на пощенските програми са вградени като функции на програми, имащи друго предназначение. Така например в програмите Netscape и MS internet Explorer, предназначени да осъществяват достъп до WWW сървъри, са интегрирани всички разгледани по-горе възможности на пощенските програми.

Протоколи за електронна поща SMTP и POP

Протоколите, които най-често се използуват за тази услуга в Интернет, са SMTP (Simple Mail Transfer Protocol), POP (Post Office Protocol) и MIME (Multipurpose Internet Mail extensions).

Прост протокол за прехвърляне на поща - SMTP

SMTP е протокол разработен за ползване от програмите за работа с електронната поща. Чрез този протокол електронната поща се предава в Интернет, в друга мрежа, която не използува TCP/IP протоколния стек, или в локалната мрежа. SMTP протоколът ползва TCP транспортния протокол, а номерът на TCP порта за него е 25.

Протоколът SMTP определя начина, по който пощата се доставя от изпращача до получателя и дефинира вътрешния формат на пощенските съобщения. Неговата функция е да осигури обмен на съобщения между два SMTP процеса (две програми). SMTP процесът, който изпраща електронна поща, се нарича SMTP клиент, а SMTP процесът, който я получава, се нарича SMTP сървър или пощенски

сървер. При работа с такъв сървър е необходимо да се използва и друг протокол (обикновено POP), за да се прехвърли пощата от сървъра на персоналния компютър.

Диалогът по протокола SMTP включва три основни етапа:

1. Установяване на връзка между SMTP клиента и SMTP сървъра, когато има писмо за изпращане. SMTP клиентът отваря TCP връзка към SMTP сървъра на хоста получател, използувайки порт 25. Клиентът изпраща команда **"Hello",** съдържаща името на изпращача. Сървърът връща отговор, в който указва възможността за приемане на писмото.
2. Предаване на поща по установената връзка. В случай, че сървърът може да приеме писмото, клиентът изпраща команда **"Mail from",** съдържаща името на изпращача. След това той издава една или повече команди **"Rcpt to",** които идентифицират получателите на писмото. Предаването на самото писмо се извършва от клиента чрез командата **"Data".** На всяка команда сървърът изпраща съответен отговор. Ако сърверът получи успешно писмото от даден потребител, клиентът го изтрива от списъка на получателите. Когато целият списък е изчерпан, както писмото, така и списъкът на получателите се изтриват от опашката със писма на изпращача. Когато обменът приключи, TCP връзката може да бъде използувана за прехвърляне на друго писмо. Ако няма повече писма за изпращане, връзката може да бъде затворена.
3. Затваряне на връзката. Клиентът затваря връзката чрез командата **"Quit".** Двете страни изпълняват TCP операцията "Close", с което връзката се прекъсва.

SMTP сървъра съхранява получените писма в своята опашка, след което ги предава към съответния пощенски сървър, на който се намира пощенската кутия на получателя на конкретното писмо. Когато това предаване е успешно, писмото се трие от опашката. В противен случай то се запазва и се правят няколко опита за повторното му предаване. Ако те не са успешни, писмото се връща на неговия подател със съобщение за грешка. Когато даден SMTP сървър получи от друг пощенски сървър писмо за получател, имащ пощенска кутия на първия сървер, то това писмо се записва в тази кутия. Въпреки че протоколът SMTP гарантира получаване на писмото в отдалечения хост, той не осигурява процедура, която да се гарантира прочитането му от потребителя, за когото то е предназначено.

Основните команди на SMTP протокола са следните:

- **Hello** - идентифицира изпращащия писмото SMTP хост, който може да бъде сървър или клиент;
- **Mail from** - идентифицира адреса на изпращача;
- **Rcpt to** - идентифицира адреса на получателя;
- **Data** - започва предаване на писмо;
- **Rset** - прекъсва предаването на писмо;
- **Help** - запитване за помощ;
- **Quit** - край на SMTP сесията.

Протоколът SMTP е описан в EFC 821.

Протокол за електронна поща - POP

Протоколът POP се използува, за да се прехвърли съдържанието на пощенската кутия на потребителя от пощенския сървер към системата на потребителя. Има две разпространени версии на този протокол - POP2 и POP3. Двете версии реализират подобни функции, но са несъвместими, тъй като използуват различни команди. Функциите на POP протоколите са прости - потребителят може да изпълни telnet към хоста на пощенския сървер, използувайки порт 109 за POP2 или порт 110 за POP3 и да задава съответните POP команди, работейки с telnet.

POP2 протоколът проверява името и паролата на потребителя. За целта клиентът издава командата HELLO, следвана от името и паролата на пощенската кутия. В отговор сърверът изпраща броят на съобщенията в пощенската кутия. След това клиентът може да издаде командата READ, с която започва четене на писмата от пощенската кутия. За да се получи пълният текст на писмото се издава команда RETR, чрез която писмото се прехвърля от сървъра към системата на потребителя. Там писмото може да се прочете чрез съответна програма за четене на електронна поща. Получаването на писмото се потвърждава с командата ACKD към сървъра, който го изтрива от пощенската кутия. След всяко потвърждение сърверът изпраща броя байтове на следващото писмо. Ако този брой е равен на нула, това означава, че няма повече писма за прехвърляне и сесията може да приключи. Това става посредством командата QUIT, която се издава от клиента.

Въпреки, че версията POP3 реализира подобни функции, нейните команди са различни. Например вместо една команда HELLO, както е при POP2, при POP3 се използуват две команди - USER. (следвана от името на потребителя), и PASS (следвана от паролата). Вместо командата REAAD при POP2, POP3 издава командата STAT, която показва броя на непрочетени съобщения и общата им дължина в байтове. В командата RETR за POP3 трябва да се зададе номер на

писмото, а за неговото изтриване се издава отделна команда DELE. Сесията приключва със същата команда - QUIT.
POP2 е описан в RHC 937, а POP3 - в RFC 1725.

А ето и диалога по протокола SMTP:
power**$telnet amigo 25**
Trying 194.141.0.3... Connected to amigo.acad.bg. Escape character is ' ^]'.
220 amigo.acad.bg ESMTP Sendmail 8.12.9/8.12,9; Mon, 05 May 2003 16:10:55 -200 (EET DST) **helo power**
250 amigo.acad.bg Hello power, pleased to meet you
help
214-Commands:
214- HELO MAIL RCPT DATA RSET
214- NOOP QUIT HELP VRFY EXPN
214-For more info use "HELP <topic>".
214-smtp
214-For local information contact postmaster at this site.
214 End of HELP info
MAIL FROM: nina@acad.bg
250 nina@acad.bg... Sender ok
RCPT TO:iavor@acad.bg
250 iavor@acad.bg... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
hello, this is a test for the book
250 Ok **QUIT**
221 amigo.acad.bg closing connection Connection closed by foreign host.

За да бъде в състояние даден хост да предлага услугата "електронна поща", е необходимо той да е част от глобална мрежа. На него трябва да се инсталира пощенски сървер.
Реализации на SMTP сървъри:
- Sendmail - разработена още в зорите на Интернет, била е почти "killer app" за тогавашния Интернет
- qmail - основна идея: използване на истинската Unix философия: малки инструменти, всяко от които прави едно нещо и го прави добре. Парична награда за пробиви в сигурността - досега не е имало.
- MS OutLook - Проблеми: голям, тежък, не толкова лесно следим (logs и т.н.)

• ssmtp, nullmailer - идеални решения за потребителска машина, вързана
към ISP, и имаща право да ползва неговия SMTP сървър

ПОЩАТА С ПОВИШЕНА ПОВЕРИТЕЛНОСТ (PEM)

PGP и други подходи за криптиране с публичен ключ попадат в категорията на *Поща с повишена поверителност (PEM),* която представлява серии от автентичност на съобщението и криптиращи процедури, създадена от Изследователския екип за поверителност и безопасност (PSRG) към Специалното звено за интернет проучвания (IRTF) и PEM работната група (PEM WG) към Специалното Интернет инженерно звено (IETF).

Стандартите на PEM позволяват да се използват различни криптиращи техники за гарантиране на поверителността, автентичността и целостта на съобщението. Целостта позволява потребителят да се увери, че никой не е променял съобщението по време на предаването му. Автентичността позволява проверка на подателя, а пове-рителността позволява да запазите това съобщение в тайна от останалите, които нямат никакво отношение с него.

АВТЕНТИЧНОСТ НА ОРИГИНАЛА с PEM

RFC 1422, "Повишена поверителност на електронната поща по Интернет: част II: Сертифицирано ключово управление", дефинира схемата за автентичност на PEM, използвайки йерархична рамка на подреждане. В основата стоят сертификатите, съдържащи функции, като алгоритъм за цифрово подписване, името на съобщението, името на създалия сертификата, периода на валидност и публичния ключ, както и придружаващия алгоритъм. Този йерархичен подход осигурява значителна гаранция, че сертификатите наистина идват от оригиналния подател, чието име носи и самият сертификат. Това затруднява и представянето на лъжлив сертификат, тъй като много малко хора ще се доверят или използват сертификати, които нямат проследяваща информация.

ПОВЕРИТЕЛНОСТ НА СЪОБЩЕНИЕТО в PEM

PEM използва стандартни криптиращи алгоритми за подходяща поверителност. RFC 1423, "Повишена поверителност на електронната поща по Интернет: част III: Алгоритми, модели и идентификатори", дефинира симетричните и асиметричните алгоритми за използване в PEM ключовото управление и криптиране. Основният стандартен алгоритъм е DES (Стандарт за криптиране на данни). DES е стандарт за електронен ключ за криптиране (ECB) и за режим Криптиране-Декриптиране-Криптиране, използвайки двойка 64-битови ключове за симетрично ключово управление. За асиметричното управление се използва алгоритъм RSA.

ИНТЕГРИРАНОСТ НА ДАННИТЕ В PEM

За да осигури интегрираност на данните, PEM използва концепция, позната като из-влечение от съобщение. Тези извлечения биват RSA-MD2 и RSA-MD5, съответно за симетричен и асиметричен ключов режим. В основата си и двата алгоритъма из-ползват "съобщения" с произволна дължина, които могат да бъдат всяко едно съобщение или файл и създават 16-битова стойност. След това PEM криптира стойността с използваната в определения момент техника за ключово управление. Когато получателя получи съобщението, може да стартира извлечението и ако то генерира същата 16-битова стойност, тогава може да се твърди, че не е променяно при предаването. Използването на извлечения се е наложило поради факта, че се изчисляват бързо и е почти невъзможно да се открият две различни по съдържание съобщения, генериращи една и съща стойност.

КРАТЪК ОБЗОР НА ОСНОВНИТЕ КРИПТИРАЩИ ПРОГРАМИ

Pretty Good Privacy (PGP)

Известния PGP софтуер на Фил Зимерман използва криптиране с публичен ключ за защита на електронна поща и файлове с данни. Той ви позволява да общувате безопасно с всеки. Програмата е с изключителни възможности и е доста бърза, притежава интелигентно ключово управление и инструменти за цифрови подписи и компресиране на данни, както и сравнително лесни за използване команди. Притежава версии за MS-DOS, Unix, Windows и Macintosh.

PGP използва няколко криптиращ метода и е наричана хибридна криптираща система, защото използва четири криптиращи елемента. В

него се съдържат симетричен шифър (общ единичен ключ), наречен Международен алгоритъм за криптиране на данни (IDEA), асиметричен шифър (двойка публичен и частен ключове - или RSA, или Дифи-Хелман, в зависимост от версията), еднопосочен хеш механизъм и стандартен генератор на произволни числа.
IDEA е алгоритъм, разработен от Ascom Zurich Швейцария. Той използва 128-битов ключ и се счита за сигурен.

КРИПТИРАНЕ И ENIGMA

Потребителите на Unix могат да използват командата crypt за криптиране на данни. Алгоритъмът, който се използва, е базиран на известните през Втората световна война криптиращи устройства Enigma, разгадани от английския математик Алан Тюринг.
Документите, криптирани с тази команда, предлагат ниско ниво на защита. Тъй като криптиращият алгоритъм, използван тук, е разработен дълго преди да започне да се използва реално огромната компютърна мощ, той се оказва и изключително лесен за пробив. Така че са необходими едва няколко часа упорита работа с бърз компютър, за да се преодолее защитата му. Затова може да се използва crypt само когато се съхранява собствена информация, която обаче не е критично важна или поверителна, но никога при важни данни и документи.

ИЗПОЛЗВАНЕ НА CRIPTOAPY НА MICROSOFT

CryptAcquireContext
CryptCreateHash
CryptDecrypt
CryptDeriveKey
CryptDestroyKey
CryptEncrypt
CiyptExportKey
CryptGenKey
CryptGetHashParam
CryptGetKeyParam
CryplGetPro vParam
CryptGetUserKey
CryptHashData
CryptlmportKcy
CryptReleaseContext
CryptSetProvParam
CryptSetProvider

CryptSignHash
CryptVeriJySignature……
……... Повече за тях в MSDN

Проверка на източника на информация чрез цифров подпис

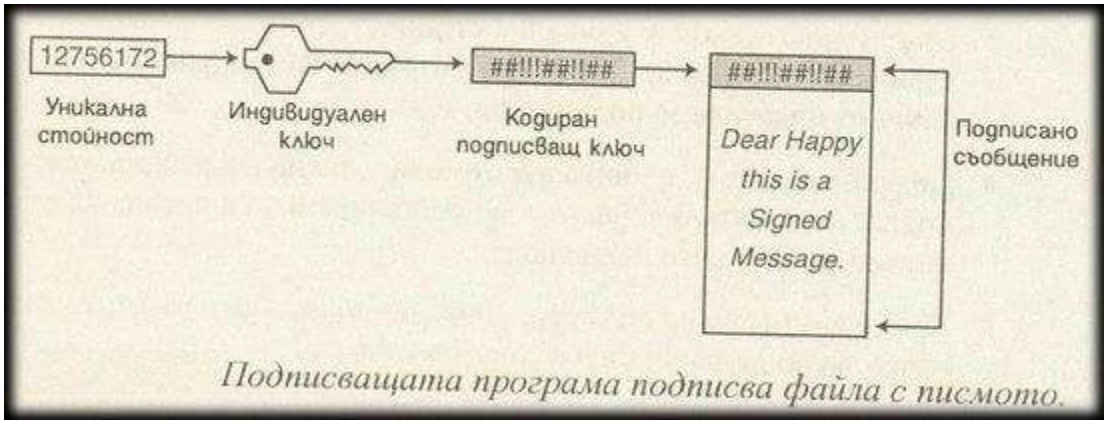
Другият основен проблем касае достоверността за документ, че именно подателят го е изпратил. С други думи, важно е всяко едно предаване да бъде достоверно. За да се гарантира, че дадено съобщение е изпратено лично от даден потребител, то трябва да бъде подписано *цифрово*.Така се добавя уникална цифрова стойност, която удостоверява, че файлът е от въпросния потребител и никой не го е променял след изпращането му.

ПРЕГЛЕД НА СЪДЪРЖАНИЕТО НА ЦИФРОВИЯ ПОДПИС

Цифровият подпис е уникална стойност, която се добавя от специално разработен за такива цели софтуер. Подписващите програми генерират подписа в две стъпки. Първо, файлът се прекарва през математическа *хеш* функция. Хеш функцията създава уникална стойност от байтовете, които са включени във файла. След това тази функция изчислява стойността така, че да не можете да извлечете файла от нея.

Файлът преминава през хеш функцията и към него се добавя уникална стойност.

След създаването на тази стойност следва криптиране с потребителския индивидуален ключ. В последния етап програмата записва подписана версия на файла, която включва информация от подписващата програма и индикатори за това, къде е началото и края на файла.



За да се удостовери подписа, получателят трябва първо да стартира софтуера за декодиране на хеш стойността на подписа чрез публичния ключ на подателя. След това тази стойност се запазва на временно място.

След това софтуерът на получателя прекарва файла през същата хеш функция, използвана и от подателя. Изчислената стойност се сравнява с тази в кодирания файл. Ако двете стойности съвпадат, програмата информира потребителя, че подписът е верен.

ИЗПОЛЗВАНЕ НА LOG ФАЙЛОВЕТЕ НА SENDMAIL ЗА ПРОСЛЕДЯВАНЕ

НА SMTP *НАПАДЕНИЯ*

Операционната система Unix записва всички транзакции на програмата sendmail във файла syslog и във log файла sendmail *(sendmal)* е програма, обслужваща комуникацията с електронна поща в Unix. И двата файла предоставят важни улики на системния администратор в случай, че хакер се опита да използва някакъв, свързан с електронната поща, бъг в операционната система чрез достъп през SMTP порта. Програмата sendmail обозначава всички регистрирани sendmail съобщения с етикет mail и ниво на критичност от debug до crit. Всички съобщения в log файла на sendmail съдържат в своя текст името на програмата.

За отбелязване е, че програмата sendmail има опция на командния ред (-L), която указва на операционната система най-ниското ниво на строгост, което ще осигури регистрирането на съобщенията в log файл. Ако искаме операционната система да записва по-подробна информация за действията на програмата, трябва да стартираме sendmail с по-висока стойност след опцията (-L). Можем да стартираме със стойност 0 (-L 0), ако не искаме запазване на информацията в log файлове.

LOG ФАЙЛ НА ПОТРЕБИТЕЛСКИТЕ КОМАНДИ

Едни от най-незабележимите log файлове, които Unix поддържа, са log файловете на потребителските команди. Тези файлове съхраняват запис на командите, които потребителят е изпълнил в обвивката на Unix при предходни сесии. И двата командни интерпретатора за Unix - C shell и Korn shell - поддържат свойството да запазват потребителските команди, генерирайки файл, наречен shell history.
Стойността на променлива от средата на командния интерпретатор определя броя на командите, които файлът запазва. В C shell променливата е *$history,* а в Korn shell тя се нарича *$HISTSIZE.* Командният интерпретатор запазва командите в директориите на съответните потребители. В C shell log файлът се нарича *.history*. В Korn shell този файл по подразбиране се нарича *.sh_history.* Въпреки това, в Korn shell можете да промените името на този файл, задавайки го като стойност на променливата *$HISTFILE.* Командата history показва на екрана съдържанието на log файловете в хронологичен ред, с поредни номера. При изпълнение на командата с опция *-H,* списъкът на командите ще бъде без поредни номера.

Тези log файлове са полезни, защото повечето хакери не забелязват тяхното съществуване или забравят да ги изтриват при първоначално влизане в системата. В голяма част от случаите хакерите могат да модифицират повечето или всичките log файлове, за които вече научихте, но пренебрегвайки history файла, ви оставят пълна картина за всяка стъпка, предприета от тях във вашата система.

СРЕДСТВА ЗА НАБЛЮДЕНИЕ В Windows NT

Windows NT също поддържа средства за наблюдение, използвайки системните възможности за регистриране на събития и програмата Event Viewer. За разлика от Unix системите обаче, Windows NT изисква специфични инструкции от системния администратор, за да започнат системите за наблюдение да работят. Точно както различните log файлове в Unix могат да помогнат да бъдат проследени потребителските действия и да бъдат открити дупки в сигурността, системите за наблюдение в Windows NT помагат те да бъдат защитени. Програмата Event Viewer ви дава важна информация за инсталацията на системата и за процедурата по стартирането.

ЗАПЛАХА ОТ ВИРУСИ, ПРЕНАСЯНИ ПО E-mail

Електронната поща е една от най-големите области на интерес за системните администратори, борещи се с вирусите. Повечето хора не съзнават начина, по който вирусът се прехвърля от една машина на друга. По същество чисто текстовите електронни съобщения не представляват заплаха за пренасяне на каквито и да е вируси. Те са обикновени файлове с данни, а не изпълними програми, от което следва, че не могат да пренасят вируси (с изключение на макро вируси). Същото важи и за повечето, но не за всичките, файлове, прикрепвани към e-mail съобщения. Някои Web браузъри и други програми за онлайн услуги стартират изпълнимите файлове, изтегляни от мрежата, веднага щом пристигнат. Значително по-широко разпространено от прикачването на текстови файлове към e-mail съобщения е пренасянето на файлове, като документи на Microsoft Word, които изглеждат като прости файлове с текстова информация. Поради последния напредък, осъществен от софтуерните компании, тези файлове не са само това, което изглеждат. Почти всички файлове, създадени в един от многото налични "продуктивни офис пакети", могат да включват макроси. Повечето хора използват макроси, за да ускорят работата си чрез спестяване на еднотипни действия. Всъщност макросите са програми, вмъкнати във файл с данни. Те могат да бъдат

използвани както за добро, така и за извършване на злонамерени действия спрямо други потребители. Всеки макрос, който се саморазпространява, е макро вирус. Въпреки че болшинството от известните макро вируси са относително безвредни, те несъмнено разполагат с разрушителен потенциал. Имайки достъп до комплект от команди, които могат да бъдат използвани за пагубни цели, макро вирусите могат да изтрият достатъчно системни файлове, така че компютърът да не може повече да се стартира.
Най-добрата защита срещу вируси, разпространяващи се чрез e-mail, както и срещу вируси, пренасящи се с дискети, е антивирусния софтуер, наличен в почти всеки компютърен магазин. За защита срещу макро вируси е добре да бъдат преглеждани всички входящи документи.

1. Домашен потребител – надали някой ще се интересува от личната кореспонденция, но все пак могат да се използва криптиращ софтуер и цифрови подписи (ако домашният потребител е чувал за тях)
   Клиентите и сървърите са споменати по горе.
2. Малка фирма – почти същото. Хубаво е да се спомене че тук е желателно да има и някакъв антивирусен софтуер.
3. Голяма корпорация – наличието на защитни стени и програми е почти задължително, както и поверителната информация да се криптира и подписва.
4. Военна структура – тук един администратор не трябва да прави никакви компромиси със сигурността (не само за e-mail) негово задължение е да изгради и защити мрежата (без да използва фрази от вида “абе и така може”, “зарежи го, к’во ти пука нали работи” .....). Идеята, е че тука администраторите ще са няколко и трябва много точно да се разберат за начина и действие на работа, инструментите които ще използват за да не създават самите те предпоставки за пробив в сигурността. Когато към техният екип се присъедини нов специалист тяхно задължение е да го въведат във цялата система и едва когато са 100% убедени че е напълно готов да го оставят да работи без постоянен контрол. И понеже за военните винаги има средства то ще бъде престъпно безотговорно ако не се закупят различни хардуерни устройства за следене и филтриране на трафика. Най-сигурния софтуер за криптиране и подписване на документи е задължителен. А ако комерсиалните клиентите и сървъри не

отговарят на нужното ниво на сигурност могат да създадат свои такива.

Заключение:

Едно e-mail съобщение преди да пристигне до крайният хост може да мине през няколко междинни. Тези междинни хостове могат да имат сериозни пропуски в своята сигурност. За това е най-добре да го криптираме и подпишем за да бъдем сигурни че дори и някой злонамерен човек да го "залови" ще му е почти невъзможно да го разбере. Факта че писмата се пазят в отделен файл *(mail box)* на хоста, както и че SMTP използва TCP транспортния протокол означава че нашето съобщение може да бъде прихванато и прочетено (разбиване на TCP сесия). Ето защо е важно да се използва специализиран софтуер за криптиране и електронен подпис използващ надеждни алгоритми (RSA, MD5).

Важно е да се отбележи че сигурността и защитата на данните е процес над който трябва непрекъснато да се работи (да се следи за излизането на нови по-надеждни протоколи, както и криптиращ софтуер), а не нещо което се прави веднъж и толкова. Хората работещи в тази област трябва непрекъснато да се "upgrejdvat", защото "лошите хора" не спят!!!

Ползвана литература:

1. Боянов К, Турлаков Х, Тодоров Д, Боянов Л, Димитров В, Желязков В. *Принципи на работа на компютърните мрежи. Интернет.*
2. Кландер Л. *Защита от Хакери... и най-добрите хакерски трикове и техники.*
3. Сайта по мрежова сигурност *http://nedyalkov.com/security* и др.