Securitatea reţelelor. Viermele Internetului

Internet este o structură deschisă, la care se poate conecta un număr mare de calculatoare fiind deci greu de controlat. De aceea putem vorbi de vulnerabilitatea reţelelor manifestată pe variate planuri. Un aspect crucial al reţelelor de calculatoare, în special al comunicaţiilor prin Internet îl constituie securitatea informaţiilor.

Categorii de atacuri asupra reţelelor

Ameninţările la adresa securităţii unei reţele de calculatoare pot avea următoarele origini: dezastre sau calamităţi naturale, defectări ale echipamentelor, greşeli umane de operare sau manipulare, fraude. Primele trei tipuri de ameninţări sunt accidentale, în timp ce ultima este intenţionată. Câteva studii de securitate a calculatoarelor estimează că jumătate din costurile implicate de incidente sunt datorate acţiunilor voit distructive, un sfert dezastrelor accidentale şi un sfert greşelilor umane. Acestea din urmă pot fi evitate sau, în cele din urmă, reparate printr-o mai bună aplicare a regulilor de securitate (salvări regulate de date, discuri oglindite, limitarea drepturilor de acces).

În ameninţările datorate acţiunilor voite, se disting două categorii principale de atacuri: pasive şi active.

1) Atacuri pasive - sunt acelea în cadrul cărora intrusul observă informaţia ce trece -prin "canal", fără să interfereze cu fluxul sau conţinutul mesajelor. Ca urmare, se face doar analiza traficului, prin citirea identităţii părţilor care comunică şi "învăţând" lungimea şi frecvenţa mesajelor vehiculate pe un anumit canal logic, chiar dacă conţinutul acestora este neinteligibil. Atacurile pasive au următoarele caracteristici comune:

- **Nu cauzează pagube (nu se şterg sau se modifică date);**
- **Încalcă regulile de confidenţialitate;**
- **Obiectivul este de a "asculta" datele schimbate prin reţea; .**

- **Pot fi realizate printr-o varietate de metode, cum ar fi supravegherea legăturilor telefonice sau radio, exploatarea radiaţiilor electromagnetice emise, rutarea datelor prin noduri adiţionale mai puţin protejate.**

2) Atacuri active - sunt acelea în care intrusul se angajează fie în furtul mesajelor, fie -in modificarea, reluarea sau inserarea de mesaje false. Aceasta înseamnă că el _ poate şterge, întârzia sau modifica mesaje, poate să facă inserarea unor mesaje â false sau vechi, poate schimba ordinea mesajelor, fie pe o anumită direcţie, fie pe -ambele direcţii ale unui canal logic. Aceste atacuri sunt serioase deoarece modifică starea sistemelor de calcul, a datelor sau a sistemelor de comunicaţii. Există . următoarele tipuri de ameninţări active:

- **Mascarada - este un tip de atac în care o entitate pretinde a fi o altă entitate. De exemplu, un utilizator încearcă să se substitue altuia sau un serviciu pretinde a fi un alt serviciu, în intenţia de a lua date secrete (numărul cărţii de credit, parola sau cheia algoritmului de criptare). O "mascaradă" este însoţită, de regulă, de o altă ameninţare activă, cum ar fi înlocuirea sau modificarea mesajelor;**
- **Reluarea - se produce atunci când un mesaj sau o parte a acestuia este reluată (repetată), în intenţia de a produce un efect neautorizat. De exemplu, este posibilă reutilizarea informaţiei de autentificare a unui mesaj anterior. În conturile bancare, reluarea unităţilor de date implică dublări şi/sau alte. modificări nereale ale valorii conturilor;**
- **Modificarea mesajelor - face ca datele mesajului să fie alterate prin modificare, inserare sau ştergere. Poate fi folosită pentru a se schimba beneficiarul unui credit în transferul electronic de fonduri sau pentru a modifica valoarea acelui credit. O altă utilizare poate fi modificarea câmpului destinatar/expeditor al poştei electronice;**
- **Refuzul serviciului - se produce când o entitate nu izbuteşte să îndeplinească propria funcţie sau când face acţiuni care împiedică o altă entitate de la îndeplinirea propriei funcţii;**

- **Repudierea serviciului - se produce când o entitate refuză să recunoască un serviciu executat. Este evident că în aplicaţiile de transfer electronic de fonduri este important să se evite repudierea serviciului atât de către emiţător, cât şi de către destinatar.**

În cazul atacurilor active se înscriu şi unele programe create cu scop distructiv şi care afectează, uneori esenţial, securitatea calculatoarelor. Există o terminologie care poate fi folosită pentru a prezenta diferitele posibilităţi de atac asupra unui sistem. Acest vocabular este bine popularizat de "poveştile" despre "hackeri". Atacurile presupun, în general, fie citirea informaţiilor neautorizate, fie (în cel mai frecvent caz) distrugerea parţială sau totală a datelor sau chiar a calculatoarelor. Ce este mai grav este posibilitatea potenţială de infestare, prin reţea sau copieri de dischete, a unui mare număr de maşini. Dintre aceste programe distructive amintim următoarele:

- **Viruşii - reprezintă programe inserate în aplicaţii, care se multiplică singure în alte programe din spaţiul rezident de memorie sau de pe discuri; apoi, fie saturează complet spaţiul de memorie/disc şi blochează sistemul, fie, după un număr fixat de multiplicări, devin activi şi intră într-o fază distructivă (care este de regulă exponenţială);**
- **Bomba software - este o procedură sau parte de cod inclusă intr-o aplicaţie "normală", care este activată de un eveniment predefinit. Autorul bombei anunţă evenimentul, lăsând-o să "explodeze", adică să facă acţiunile distructive programate;**
- **Viermii - au efecte similare cu cele ale bombelor şi viruşilor. Principala diferenţă este aceea că nu rezidă la o locaţie fixă sau nu se duplică singuri. Se mută în permanenţă, ceea ce îi face dificil de detectat. Cel mai renumit exemplu este Viermele INTERNET ului, care a scos din funcţiune o parte din INTERNET în noiembrie 1988;**
- **Trapele - reprezintă accese speciale la sistem, care sunt rezervate în mod normal pentru proceduri de încărcare de la distanţă, întreţinere sau pentru dezvoltatorii unor aplicaţii. Ele**

permit însă accesul la sistem, eludând procedurile de identificare uzuale;

- **Calul Troian - este o aplicaţie care are o funcţie de utilizare foarte cunoscută şi care, intr-un mod ascuns, îndeplineşte şi o altă funcţie. Nu creează copii. De exemplu, un hacker poate înlocui codul unui program normal de control "login" prin alt cod, care face acelaşi lucru, dar, adiţional, copiază într-un fişier numele şi parola pe care utilizatorul le tastează în procesul de autentificare. Ulterior, folosind acest fişier, hacker-ul va penetra foarte uşor sistemul.**

Modelul de securitate în reţele

Modelul de securitate pentru un calculator seamănă cu o ceapă. Niveluri de securitate înconjoară subiectul ce trebuie protejat. Fiecare nivel izolează subiectul şi îl face mai greu de accesat în alt mod decât în cel în care a fost planificat.

1) **Securitatea fizică reprezintă nivelul exterior al modelului de securitate şi constă, în general, în încuierea echipamentelor informatice într-un birou sau într-o altă incintă. Securitatea fizică merită o consideraţie specială. Problema cea mai mare o constituie salvările pentru copii de rezervă ale datelor şi programelor şi siguranţa păstrării suporţilor de salvare. În aceste situaţii, reţelele locale sunt de mare ajutor: dacă toate fişierele schimbate frecvent rezidă pe un server, aceleaşi persoane (sigure şi de încredere), care lansează salvările pentru mainframe-uri, pot face acelaşi lucru şi la server. Calculatorul, ca orice piesă costisitoare, ar trebui să fie protejat şi. de pericolul furtului. Păstrarea în afara zonelor publice este una dintre cele mai bune forme de protecţie. Simpla încuiere a echipamentelor va preveni mutările ascunse precum şi furtul. Într-un sistem în care prelucrarea este distribuită, prima măsură de securitate fizică care trebuie avută în vedere este prevenirea accesului la echipamente. Pentru a învinge orice alte măsuri de securitate, trebuie să se dispună de acces fizic la**

echipamente. Acest lucru este comun tuturor sistemelor de calcul, distribuite sau nu.

2) **Securitatea logică constă din acele metode care asigură controlul accesului Ia resursele şi serviciile sistemului. Ea are, la rândul ei, mai multe niveluri, împărţite în două grupe mari: niveluri de securitate a accesului (SA) şi niveluri de securitate a serviciilor (SS).**

3)

- **Securitatea accesului (SA) cuprinde:**
  - **accesul la sistem (AS), care este răspunzător de a determina dacă şi când reţeaua este accesibilă utilizatorilor. EI poate fi, de asemenea, răspunzător pentru decuplarea unei staţii, ca şi de gestiunea evidenţei accesului. AS execută, de asemenea, deconectarea forţată, dictată de supervizor. AS poate, de exemplu, să prevină conectarea în afara orelor de serviciu şi să întrerupă toate sesiunile, după un anumit timp;**
  - **accesul la cont (AC), care verifică dacă utilizatorul care se conectează cu un anumit nume şi cu o parolă există şi are un profil utilizator valid;**
  - **drepturile de acces (DA), care determină ce privilegii de conectare are utilizatorul (de exemplu, contul poate avea sesiuni care totalizează 4 ore pe zi sau contul poate utiliza doar staţia 27).**

- **Securitatea serviciilor (SS), care se află sub SA, controlează accesul la serviciile sistem, cum ar fi fire de aşteptare, I/O la disc şi gestiunea server ului. Din acest nivel fac parte:**
  - **controlul serviciilor (CS), care este responsabil cu funcţiile de avertizare şi de raportare a stării serviciilor; de asemenea, el activează şi dezactivează diferitele servicii;**
  - **drepturile la servicii (DS), care determină exact cum foloseşte un anumit cont un serviciu dat; de exemplu, un cont poate avea numai dreptul de a adăuga fişiere la spooler-ul unei imprimante, dar are drepturi depline, de a adăuga şi şterge fişiere, pentru o altă imprimantă.**

O dată stabilită conexiunea, SA validează şi defineşte contul. Operaţiile ce trebuie executate sunt controlate de SS, care împiedică cererile ce nu sunt specificate în profilul utilizatorului. Accesul intr-un sistem de securitate perfect trebuie să se facă prin aceste niveluri de securitate, de sus în jos. Orice sistem care vă lasă să evitaţi unul sau mai multe niveluri ale modelului de securitate implică riscul de a fi nesigur.

Viermele Internet-ului

Introducere

În seara zilei de 2 noiembrie 1988, după ora 17, un program ciudat era executat pe mai multe dintre calculatoarele Internet. Acest program aduna informaţii despre host-uri, reţele şi utilizatori şi folosea aceste informaţii pentru a stabili conexiuni reţea şi pentru a pătrunde pe alte maşini. El folosea anumite defecte sau slăbiciuni prezente în anumite programe. După această pătrundere, programul se multiplica, iar copia sa încerca infectarea altor sisteme, în aceeaşi manieră. Chiar dacă programul nu a infectat decât sistemele Sun Microsystems Sun 3 şi calculatoarele VAX pe care rulau variante ale lui 4 BSD UNIX, el s-a extins rapid, creând confuzie şi consternare în rândul administratorilor de sistem şi al utilizatorilor, atunci când aceştia au descoperit invazia produsă în sistemele lor. Deşi se cunoştea că UNIX-ul are câteva slăbiciuni de securitate, în special în modul obişnuit de operare în medii deschise, totuşi scopul şi modul în care acestea au fost folosite a constituit o surpriză mare pentru toată lumea.

Programul era ciudat pentru utilizatori, în special din punctul de vedere al punctelor de apariţie. Au fost introduse fişiere neobişnuite în directoarele lusrltmp, iar unele mesaje stranii au apărut în fişierele unor utilitare, cum ar fi sendmail. Totuşi, cel mai notabil efect a fost faptul că sistemele au devenit din ce în ce mai încărcate cu procese datorită infectării multiple. Cu trecerea timpului, unele dintre maşini au devenit atât de încărcate, încât nu

au mai fost capabile să lucreze; unele maşini au fost blocate complet, atunci când spaţiul de evacuare (swaping) sau tabela de procese au fost saturate.

În dimineaţa zilei de 3 noiembrie, personalul de la Universitatea Berkeley din California şi de la Institutul de Tehnologie Massachusetts au "capturat' copii ale programului şi au început să le analizeze. Utilizatori din alte locuri au început, de asemenea, să studieze programul şi au fost dezvoltate metode de eradicare a acestuia. O caracteristică a programului a ieşit în evidenţă: modifica resursele sistemului într-un fel care nu putea fi detectat rapid. Au fost alterate fişiere şi distruse informaţiile, ceea ce a impus căutarea unei soluţii. După ora 5 am la mai puţin de 13 ore de la prima descoperire a programului, Computer Systems Research Group de la Berkeley a stabilit un set provizoriu de măsuri, în vederea opririi extinderii. Printre acestea se afla şi o modificare la serviciul sendmail şi sugestia de a redenumi compilatoarele C şi încărcătorul, pentru a preveni utilizarea lor. Aceste sugestii au fost publicate în listele de poştă electronică şi prin sistemul de ştiri din reţeaua Usenet, cu toate că extinderea viermelui a fost împiedicată, cel mai adesea prin deconetarea sistemelor de la INTERNET, în încercarea de a le dezinfecta ulterior.

Pe la orele 9 pm, în aceeaşi zi, a fost descoperită şi publicată la Purdue University o altă metodă simplă pentru stoparea invaziei acestui program. Îmbunătăţiri soft au fost anunţate şi de către grupul Berkeley, pentru a "astupa" fisurile ce permiteau programului să invadeze sistemele. Tot ceea ce rămânea de făcut era de a se analiza codul care a generat aceste probleme şi de a descoperi cine şi de ce lansat viermele.

Cronologia evenimentelor

Este foarte interesant de remarcat viteza şi profunzimea cu care Viermele s-a extins şi este semnificativ de urmărit rapiditatea cu care a fost identificat şi-a oprit utilizându-se aceeaşi reţea pentru comunicarea între specialişti a rezultatelor. Este, credem

noi, foarte interesant şi instructiv de urmărit desfăşurarea în timp a răspândirii Viermelui, cel mai important eveniment de securitate din istoria INTERNET-ului, dar -este la fel de util de a vedea rapiditatea cu care s-a cristalizat riposta specialiştilor.

2 Noiembrie 1988

- **17.00. Viermele este executat pe o maşină la Cornell University;**
- **18.00. Maşina prep.ai.mit.edu a lui MIT a fost infectată. Prep era o maşină cu acces public, utilizată pentru stocarea şi distribuirea soft-ului prin proiectul GNU. Maşina a fost configurată cu câteva vulnerabilităţi de securitate de notorietate, care permiteau . utilizatorilor de la distanţă să introducă fişiere în sistem;**
- **18.30. Maşina infectată de la Pittsburgh Univerity a infectat o maşină a corporaţiei RAND;**
- **21.00. Viermele este descoperit pe maşinile de la Stanford University; 21.30. Este invadată o primă maşină la Minnesota University;**
- **21.34. Maşina gateway a Universităţii Berkeley din California este invadată. Se descoperă o neobişnuită incărcare a maşinii cu procese de poştă;**
- **22.34. Este infectată maşina gateway a Universitţăii Princeton;**
- **22.40. Maşini de la Universitatea Norih Carolina sunt infectate şi sunt încercări de a invada alte maşini;**
- **22.48. Sunt infectate maşini ale SRI via sendmail (poştă);**
- **22.52. Viemele incearcă să invadeze maşina andrew.cmu.edu de la Universitatea Carnegie-Mellon (poştă);**
- **22.54. Calculatoarele gateway de la Universitatea din MaryJand sunt atacate prin . . procesul din fundal corespunzător programului fingerd;**
- **22.59. Maşinile de la Universitatea din Pennsylvania sunt atacate, dar sunt "insen-sibile". Vor fi depistate 210 încercări de infectare în următoarele 13 ore, prin poştă; 23.48. Calculatorul mimsy.umd.edu de la Universitatea din Maryland este infectat via sendmail (poştă);**

- **23.40. Cercetătorii de la Berkeley descoperă ţintele de atac ca fiind sendmail şi rsh. Ei incep să închidă serviciile pentru alte reţele, ca măsură de precauţie;**
- **23.45. Maşinile de la Dartmouth şi Laboratorul de Cercetări Balistice al Armatei (BRL) sunt atacate şi infectate (poştă, NCSC);**
- **23.49. Gateway-urile de la Universitatea din Utah sunt infectate. În următoarea oră numărul încercărilor va ajunge la 100;**

3 Noiembrie 1988

- **00.07. Este infectată maşina Univesităţii din Arizona, prin poştă;**
- **00.21. Este infectată maşina principală a Universităţii Princeton (un VAX 8650). Numărul încercărilor ajunge la 68 şi maşina clachează;**
- **00.33. Este infectată maşina dewey.udel.edu a Universităţii din Delaware;**
- **01.30. Maşinile de la UCLA sunt infectate;**
- **02.00. Viermele este identificat pe maşinile de la Universitatea Harvard;**
- **02.38. De la Berkeley se transmite un mesaj prin poştă cu conţinutul: "Suntem atacaţi". Domeniile menţionate ca fiind infectate sunt: U.C.Berkeley, U.C.San Diego, LLL, Stanford şi NASA Ames;**
- **03.15. Sunt infectate maşiniie de la Universitatea din Chicago. Una dintre maşinile de la Departamentuf de fizică suferă 225 de încercări de infectare, via fingerd, de la maşini din Cornell;**
- **03.39. Avertismentul despre Vierme este transmis de la foo@bar.arpa sub forma: "Este probabil un virus pierdut prin INTERNET.". Urmau trei scurte fraze despre cum să fie oprit Viermele, urmate de "Sper că acestea ajută, dar mai mult, sper că este vorba de o farsă.". Cel ce transmitea s-a dovedit a fi Andy Sudduth de fG Narvard, care a fost sunat prin telefon de presupusul autor al Viermelui, Robert T Morris. Datorită**

încărcării reţelei şi maşinilor, avertismentul nu este propagat ir următoarele 24 de ore;

- **04.00. Universitatea Colorado este şi ea supusă atacului; 04.00. Maşinile de la Universitatea Purdue sunt infectate;**
- **05.54. Se transmite prin poştă un avertisment cu privire la Virme şi, în plus, c măsură de protecţie minimală referitoare la programul sendmail. Mesajul său este preluat de grupul de ştiri Usenix;**
- **06.45. Se sună la National Computer Security Center şi se informează despre Vierme;**
- **07.00. Maşini ale Institutului de Tehnologie din Georgia sunt infectate. Maşina gateway (un VAX 780) suferă peste 30 de încercări;**
- **07.30. Se descoperă infectarea maşinilor de la Universitatea Purdue. Maşinile sun atât de încărcate, încât nu se puteau citi mesajele primite prin poştă, inclusiv mesajul despre Vierme;**
- **08.07. La Berkeley se identifică atacul Viermelui prin intermediul programului fingerd, dar mesajul trimis prin poştă nu poate fi citit mai bine de 13 ore;**
- **08.18. Se retransmite avertismentul despre Vierme grupului de ştiri Usene news.announce.important şi altor 30 de site-uri. Acestea au fost primele informaţii despre Vierme, aflate de cei vizaţi în cursul întregii zile, acest grup a schimb de mesaje prin poştă cu privire la progresul şi comportarea Viermelui;**
- **10.36. Se transmite prima descriere cu privire la modul de lucru al Viermelui cele din lista nntp-managers. Atacul prin programul fingerd la această oră încă nu est cunoscut;**
- **11.30. Defense Communications Agency inhibă bridge-urile de poştă între Arpane şi Milnei;**
- **13.00. Sunt blocate peste 130 de maşini ale SRI;'**
- **14.50. Personalul de la Purdue descoperă maşini infectate cu variante noi de programe sendmail instalate. Se transmite un mesaj prin poştă referitor la faptul c noua versiune de sendmail nu constituie o măsură de protecţie suficientă. Acel lucru era cunoscut deja în multe locuri, inclusiv la Berkeley şi MIT de mai bine câte ore, dar nu se publicase încă nimic;**

- **16.00. Administratorii de sisteme de la Purdue se întâlnesc pentru a stabili strateg locală. Versiunile de Vierme capturate au furnizat o variantă de prevenire a infecţie prin crearea unui director cu numele sh în directorul lusr tmp;**
- **18.00. La Purdue s-a descoperit cum lucrează Virmele, cu defecţiunea din program finger,**
- **19.00. La MIT, s-a reconstituit atacul Viermelui prin intermediul programului fingerd' şi s-a telefonat la Berkeley pentru a se anunţa aceasta. Nu a fost transmis nimic prin poştă despre acest mod de atac;**
- **19.19. S-au transmis noile îmbunătăţiri aduse programelor sendmail şi fingerd, dar aceste mesaje au fost recepţionate abia a doua zi;**
- **19.37. De la Universitatea din Rochester a fost trimisă prin poştă o descriere a atacului prin intermediul programului fingerd;**
- **21.30. Grupul de la Berkeley începe decompilarea Viermelui, pentru a determina sursa în C.**

4 Noiembrie 1988

- **00.50. Se trimite prin poştă o descriere a atacului prin intermediul fingerd. Se fac şi primele comentarii referitoare la stilul de cod al autorului Viermelui;**
- **05.00. Grupul MiT încheie decompilarea codului;**
- **09.00. Grupul de la Berkeley încheie decompilarea codului;**
- **11.00. Sunt reinstalate bridge-urile de poştă între Milnet-Arpanet;**
- **14.20. Se retransmit prin poştă modificările aduse la programul fingerd;**
- **15.36. De la MIT, se transmit clarificări asupra modului de operare a Viermelui; · 17.2d. Se transmite un set final de îmbunătăţiri pentru sendmail şi fingerd;**
- **21.30. Autorul Viermelui este identificat din două surse independente ca fiind Robert T Morris, fiut directorului ştiinţific al Centrului Naţional de Securitate a Calcutatoarelor (GNSC), Robert Morris.**

- **Până pe 8 noiembrie, marea majoritate a maşinilor au fost reconectate la INTERNET şi traficul a revenit la normal. În aceeaşi dimineaţă, aproximativ 50 de cercetători s-au intâlnit cu oficialităti din Centru! National de Securitate. Cu această ocazie, au fost identificate direciiile ulterioare de acţiune în acest domeniu. Analizatorii de trafic al reţelei au continuat să identifice încercări de infectare încă existente pe maşinile INTERNET-ului. O ultimă încecare a fost identificată la începutul lunii decembrie 1988.**

Despre autorul Viermelui

După ce Viermele a fost oprit, au fost puse, inevitabil, două întrebări: "cine?" şi "de ce?".

La prima întrebare răspunsul a apărut rapid prin identificarea lui Robert T. Morris de către New York Times. Există multe elemente care susţin identificarea făcută. Multe oficialităţi federale au afirmat că au dovezi, obţinute de la persoane distincte, prin care se specifică faptul că Morris a discutat cu aceste persoane despre Vierme şi cercetările sale în această direcţie. Ei susţin, de asemenea, că au înregistrări de pe calculatoarele de la Universitatea Cornell reprezentând versiuni de început ale codului Viermelui testate pe maşini din campus şi, de asemenea, susţin că au copii ale Viermetui găsite în contul lui Morris. Raportut furnizat de Oficiul Rectoratului Universităţii din Cornelt îl indică de asemenea pe Morris ca fiind culpabil şi prezintă motive convingătoare pentru a susţine această concluzie.

Dar dacă autorul era stabilit, motivul acestei acţiuni rămânea neclar, plasat între un experiment greşit şi până la un act inconştient de răzbunare a lui Morris împotriva tatălui său. Din studiul făcut de multe persoane asupra codului decompilat, au rezultat două concluzii:

O primă concluzie se referă la faptul că programul nu conţine, în mod explicit, porţiuni de cod care ar provoca explicit distrugeri ale sistemelor pe care ar rula. Luând în considerare abilitatea şi cunoştinţele evidenţiate de cod, pentru autor ar fi

constituit o chestiune simplă introducerea unor astfel de comenzi, dacă aceasta ar fi fost intenţia lui. În cele din urmă, eliberarea prematură în reţea a Viermelui arată că intenţia autorului de a distruge sau perturba structuri şi sisteme nu poate fi luată în considerare în mod explicit;

A doua concluzie se referă fa faptul că în cod nu este inclus un mecanism pentru a opri dezvoltarea Viermelui. Luând în considerare acest lucru, precum şi complexitatea şirului utilizat ca argument, necesar pentru a declanşa Viermele, multe persoane care au examinat codul nu consideră că Viermele a fost declanşat accidental sau că intenţia a fost de a nu fi propagat puternic. Având în vedere aceste lucruri, sunt ciudate încercările făcute pentru a justifica acţiunea lui Morris, susţinându-se că intenţia lui era de a demonstra ceva despre securitatea 7NTERNET ului sau că a fost un experiment nevinovat. Raportul Rectoratului Universităţii din Cornell nu încearcă să scuze comportamentul lui Morris. Această acţiune este etichetată ca fiind neetică şi contrară standardelor profesionale. Acţiunea sa este considerată a fi îndreptată împotriva politicii Universităţii şi practicii acceptate şi ar fi fost de aşteptat ca, având în vedere experienţa pe care o are în acest domeniu, să cunoască că astfel de acţiuni sunt nepermise. Cei care cred că Viermele constituie un accident sau un experiment nefericit sunt de părere ca autorul să nu fie pedepsit, mergând până la a cere pedepsirea administratorilor şi operatorilor de pe sistemele şi maşinile afectate, pentru neglijenţa cu care au tratat aspectele ,de securitate. Ceilalţi consideră că autorul trebuie să fie pedepsit sever, inclusiv cu privarea de libertate. Comisia de la Cornell a recomandat unele pedepse, dar nu atât de severe încât să afecteze cariera ulterioară a lui Morris. În acea recomandare este specificată suspendarea lui Morris din Universitate pentru minimum un an. Faptul că nu s-au întâmplat mari nenorociri poate constitui un accident şi este posibil ca intenţia autorului să fi fost de a supraîncărca lNTERNET-ul, aşa cum s-a şi întâmplat. Scuzarea unor astfel de acte de vandalism, sub declaraţia că autorii nu au vrut să creeze mari neajunsuri, nu poate conduce la

descurajarea repetării unor astfel de încercări, ba chiar mai mult, acestea sunt încurajate.

Vulnerabilităţi exploatate de Vierme

Viermele utilizează o serie de defecte sau slăbiciuni existente în software-ul standard al multor sisteme UNIX. Unele dintre aceste defecte sunt descrise în continuare.

Programul Fingerd

Programul fingerd este un utilitar care permite obţinerea de informaţii despre utilizatori. De obicei, este folosit pentru a identifica numele întreg sau numele de conectare (login) al unui utilizator, dacă acesta se află în sesiune şi posibil, alte informaţii despre persoana respectivă, cum ar fi numerele de telefon etc. Acest program este rulat ca daemon sau proces în fundal (background), pentru rezolvarea cererilor de informaţii venite de la distanţă, utilizându-se protocolul fingerd, Acest program acceptă conexiuni de. la programe ce rulează în altă parte, citeşte linia de intrare şi trimite răspuns receptorului care a adresat întrebarea.

Punctul slab exploatat, prin care se "sparge" acest program, implică modificarea buffer ului de intrare folosit de acesta. Biblioteca f/0 a limbajului C are câteva rutine care citesc intrarea fără a verifica limitele buffer-ului implicat în această operaţiune. În particular, apelul funcţiei gets preia datele de intrare într-un buffer, fără a face verificarea limitelor acestuia. Apelul acestei funcţii a fost exploatat de Vierme. Rutina gets nu este singura care are acest neajuns. O întreagă familie de rutine din biblioteca C-ului face posibilă depăşirea buffer ului, atunci când se decodifică intrarea sau când se formatează ieşirea, dacă utilizatorul nu specifică explicit numărul de caractere pentru conversie.

Cu toate că programatorii experimentaţi sunt cunoscători ai acestor probleme, mulţi dintre ei continuă să folosească aceste rutine. Necazul este că orice server de reţea sau program privilegiat, care utilizează aceste funcţii, poate fi compromis

datorită utilizării unei intrări improprii. Interesant este că recent, au mai fost descoperite încă două comenzi în standardul BSD UNIX, care au această problemă.

După atacul asupra INTERNET-ului au fost relevate mai multe probleme potenţiale şi mai multe modalităţi de a le înlătura, dar cu toate acestea, biblioteca cu aceste rutine continuă să fie utilizată.

Programul Sendmail

Programul sendmail este un serviciu de poştă electronică, destinat să ruteze scrisorile într-o reţea eterogenă. Programul are mai multe moduri de operare, dar unul dintre acestea este exploatat de Vierme şi implică lansarea serviciului ca proces în background (daemon). În acest mod de lucru, procesul se află în starea de "ascultare" la un port TCP (25), pentru a face distribuirea poştei sosite prin protocolul standard INTERNET, SMTP (Simple Mail Transfer Protocon. Când o astfel de situaţie este detectată, procesul intră într-un dialog cu un alt proces de la distanţă, pentru a determina expeditorul, destinatarul, instrucţiunile de distribuire şi conţinutul mesajului.

Punctul slab exploatat în sendmail este legat de o opţiune de depanare a codului. Viermele transmite comanda DEBUG la sendmail şi apoi specifică destinatarul mesajului, ca un set de comenzi şi nu ca o adresă utilizator. Într-o operaţiune normală, acest lucru nu este permis; însă, în activitatea de depanare a codului este posibilă verificarea poştei sosite pentru un anumit destinatar, fără a se apela rutinele de adresare. Prin utilizarea acestei opţiuni, testele pot rula programe care să afişeze starea sistemului de poştă, fără trimiterea de mesaje sau stabilirea unei conexiuni. Această opţiune de depanare este adesea utilizată tocmai datorită complexităţii configurării lui sendmail.

Programul sendmail este de mare importanţă, mai ales pentru sisteme UNIX derivate din BSD, deoarece mânuieşte procese complexe de rutare şi distribuire a poştei. Totuşi, în ciuda importanţei mari şi a utilizării largi, cea mai mare parte a administratorilor de sisteme ştiu puţin despre felul în care lucrează

sendmail. Deşi sunt relatate multe apariţii de driver e scrise de administratori de sisteme sau modificări aduse Kernel-ului, nimeni nu a adus încă modificări la sendmail sau la configuraţia fişierelor sale. În concluzie, punctele slabe prezentate în sendmail sunt puţin cunoscute, iar unele dintre ele sunt depistate şi comunicate pe măsura descoperirii lor.

Parole

Una din "piesele de rezistenţă" ale Viermelui implică încercarea de a descoperi parolele utilizatorilor. În sistemele UNIX, utilizatorul furnizează o parolă ca semn de verificare a identităţii. Parola este criptată, utilizând o versiune a algoritmului DES, iar rezultatul este comparat cu rezultatul criptării anterioare, prezent în fişierul letclpasswd. Dacă acestea coincid, accesul este permis. În acest fişier nu sunt incluse parolele în clar şi algoritmul se presupune a fi neinversabil; deci, fără cunoaşterea parolei nu avem acces.

Organizarea parolelor în UNIX permite unor comenzi neprivilegiate să utilizeze informaţii din fişierul /etc/passwd şi să acceseze schema de autentificare a parolelor. Deci se permite un atac prin criptarea unei liste cu parole posibile şi compararea rezultatelor cu fişierul letclpasswd, fără a se face apel la o funcţie sistem, special dedicată. De fapt, securitatea parolelor este asigurată în principal prin numărul mare de încercări ce trebuie efectuate pentru a le determina, cu toate combinaţiile de caractere posibile. Din nefericire, există maşini care lucrează rapid şi costul unei astfel de acţiuni este în continuă descreştere, datorită rapidităţii dezvoltării produselor hard.

Divizând procesul pe mai multe procesoare, se reduce mult timpul necesar determinării unei parole. Astfel de atacuri sunt uşurate mult, atunci când utilizatorul alege drept parolă un cuvânt comun sau des folosit. În acest caz, toată căutarea se rezumă la determinarea parolei prin verificarea unor cuvinte comune, existente într-o astfel de listă (vezi capitolul 2).

Viermele utilizează pentru spargerea parolei un astfel de tip de atac. În acest sens se foloseşte o listă de cuvinte standard,

cuvinte care sunt considerate a fi parole posibile. Viermele asigură criptarea lor prin intermediul unei versiuni rapide a algoritmului de cifrare şi apoi, compară rezultatul cu conţinutul fişierului /etc/passwd. Deci Viermele exploatează accesul la acest fişier, cuplat cu tendinţa utilizatorilor de a alege cuvinte comune drept parole.

Un defect discutat în prezent şi care a fost exploatat de Vierme implică utilizarea sesiunilor de încredere. Una din facilităţile utile ale soft-ului de reţea al lui BSD UNIX este suportul de execuţie a proceselor pe maşini aflate la distanţă. Pentru a se evita repetarea tipăririi parolelor pentru accesul în conturi aflate la distanţă, se asigură posibilitatea unui utilizator de a specifica o listă cu perechi gazdă/cont, care sunt considerate a fi de încredere, în sensul că un acces la distantă de la calculatorul gazdă la acel cont se face fără a utiliza parola contului respectiv. Acest aspect este responsabil de numeroasele accese neautorizate la calculatoare, dar continuă să fie utilizat, fiind convenabil. Viermele a exploatat acest mecanism prin încercarea de a localiza host-urile de încredere şi a determina perechile corespunzătoare. Acest lucru a fost realizat prin examinarea de către Vierme a fişierului de pe host-ul curent, care conţine perechile host/conturi. Odată ce Viermele găseşte astfel de candidaţi, va încerca, în modul cel mai rapid, să se autoinstaleze pe aceste maşini, folosind facilitatea execuţiei la distanţă, copiindu-se pe sine pe maşina de la distanţă, ca şi cum ar fi un utilizator autorizat, care efectuează o operaţie standard de la distanţă. Pentru a inlătura astfel de încercări în viitor, este necesar ca actualul mecanism de acces la distanţă să fie anulat şi înlocuit cu ceva nou. Un mecanism nou creat, care se apropie de cerinţele de mai sus, este server ul de autentificare Kerberos (vezi subcapitolul 7.4).

Descrierea Viermelui

Viermele INTERNET este constituit din două părţi: un program principal şi un program vector (bootstrap).

- **Programul principal, o dată instalat pe o maşină, va colecta informaţii despre alte maşini din reţea, cu care calculatorul gazdă poate fi conectat. Va face această colectare prin citirea fişierelor de configurare şi prin lansarea proceselor corespunzătoare programelor utilitare de sistem, care furnizează informaţii despre -starea curentă a conexiunilor din reţea. Apoi, va încerca să profite de fisurile din soft, descrise mai sus, pentru a instala programul său vector pe fiecare din aceste calculatoare aflate la distanţă;**
- **Programul vector are 99 de linii de cod C, care vor fi compilate şi rulate pe maşina de la distanţă. Sursa acestui program va fi transferată la "victimă", folosind una dintre metode4e care vor fi prezentate în continuare. Apoi, sursa va fi apelată pe maşina victimă, prin intermediul unei linii de comandă, cu trei argumente: adresa de reţea a calculatorului infectat, numărul port-ului de reţea pentru conectarea la această maşină (pentru a se copia fişierele programului principal al Viermelui) şi un număr magic, care acţionează efectiv ca o parolă cu o singură utilizare.**

Dacă Viermele "server", de pe maşina de la distanţă, nu recepţionează înapoi acelaşi număr magic (cel trimis de programul vector) înainte de începerea transferului, va fi deconectat imediat de la programul vector. Acest lucru este realizat pentru a preveni încercările de capturare a fişierelor binare prin înşelarea Viermelui "server".

Codul vectorului acţionează pentru ascunderea sa şi prin punerea pe 0 a vectorului său de argumente (imaginea liniei de comandă), şi prin utilizarea mecanismului fork pentru crearea unui proces fiu identic. Dacă se întâmplă o defecţiune în timpul transferului de fişiere, codul determină ştergerea fişierelor deja transferate şi apoi, face ieşirea.

O dată instalat pe o maşină, programul vector se va conecta cu calculatorul pe care se execută procesul Vierme original şi va executa transferul unor fişiere binare (cod precompilat) către maşina locală. Fiecare fişier binar reprezintă o versiune a

programului Vierme principal, compilat corespunzător unei anumite arhitecturi de calculator şi unei anumite versiuni de sistem de operare. Programul vector va transfera, de asemenea, o copie a sursei sale folosită pentru infectarea altor sisteme. Una dintre trăsăturile mai curioase ale programului vector a generat multe întrebări: programul are alocate structuri de date pentru transferul a 20 de fişiere, din care au fost folosite doar 3. Acest lucru a condus la speculaţia că a fost planificată, probabil pentru a fi lansată la o dată ulterioară, o versiune extinsă a Viermelui, cu alte fişiere de comandă, fişiere de parole, viruşi locali şi programe de tip "cai troieni".

Odată ce fişierele binare au fost transferate, programul vector încarcă fişierele corespunzător versiunii locale a bibliotecilor standard, pe care le leagă cu programele aduse. Ulterior, aceste programe vor fi apelate. Dacă unul dintre acestea rulează cu succes, citeşte în memoria proprie copiile programului vector şi ale fişierelor binare şi apoi şterge originalele de pe disc.

În pasul următor, va încerca să forţeze pătrunderea pe alte maşini. Dacă nici una dintre versiuni nu rulează cu succes, atunci mecanismul de rulare a programului vector (un fişier de comenzi) va şterge toate fişierele de pe disc create pe durata încercării de infectare, lăsând sistemul curat.