Wlamania w systemie Linux i metody ochrony przed nimi.

Autorzy: Michal Wargacki, Lukasz Baran IVFDS

STRESZCZENIE

Praca zawiera krótka charakterystyke zródel zagrozen, opis podstawowych rodzajów i technik wlaman (socjotechnika, ataki slownikowe, ataki „brute force", ataki „buffer overflow" – explioty). Zarys ochrony przed wlamaniami: polityka bezpieczenstwa, linuxowy firewall typu „statefull inspection" – Netfilter, narzedzia do testowania firewalla – skanery portów (Nmap). Narzedzia do wykrywania wlaman zewnetrznych i wewnetrznych: Snort oraz LIDS.

SPIS TRESCI

1. WSTEP

Na przestrzeni kilku ostatnich lat daje sie zauwazyc gwaltownie rosnaca popularnosc systemu Linux. Zyskuje on coraz wieksza popularnosc i uznanie zarówno w gronie „pospolitych" uzytkowników jak równiez profesjonalistów „z branzy". Linux przestal juz byc ciekawostka dla waskiego grona zapalenców i stal sie dojrzala, liczaca sie platforma na rynku systemów operacyjnych. Potwierdzeniem tego faktu jest ciagle rosnaca liczba aplikacji zarówno powstajacych od podstaw dla Linuxa jak równiez tych kojarzonych wczesniej z innymi systemami operacyjnymi – nie koniecznie unixowymi. Coraz czesciej sa to zastosowania krytyczne z punktu widzenia bezpieczenstwa, szczególnie jesli zwrócimy uwage na fakt, ze wiekszosc systemów Linuxowych pracuje w publicznej sieci Internet i jest tym samym w zasiegu potencjalnego wlamywacza [8].

2. RODZAJE I TECHNIKI WLAMAN

2.1 Zródla zagrozen

Do najbardziej powszechnych i zarazem najniebezpieczniejszych wlaman dochodzi za posrednictwem publicznych sieci komputerowych. Jest to naturalne zwazywszy na fakt, ze Internet jest medium dostepnym w skali swiatowej. Pozwala to wlamywaczom na równie latwe atakowanie odleglych systemów, znajdujacych sie niejednokrotnie w innych krajach lub kontynentach jak równiez tych pracujacych w lokalnych sieciach metropolitarnych lub kampusowych.

2.2 Rodzaje wlaman

Niekiedy duze odleglosci geograficzne sprzyjaja tez wiekszemu poczuciu bezkarnosci i prowokuja do bardziej agresywnych zachowan, szczególnie jesli chodzi o ataki na popularne, dobrze chronione systemy. Nie oznacza to, ze tylko wazne systemy powinny byc szczególnie chronione. Czesto pierwszym krokiem wlamywacza jest zdobycie dostepu do pozornie malo interesujacych systemów takich jak domowe stacje robocze aby nastepnie wykorzystac je do

ataku na docelowy system pozostajac przy tym anonimowym na wypadek wykrycia wlamania przez administratora. Równie popularnym scenariuszem jest seria wlaman „na ilosc".

Udane wlamanie prawie zawsze ma na celu przejecie kontroli nad zdalnym systemem na dluzszy czas co w przypadku Linuxa polega na zdobyciu uprawnien root'a oraz umiejetnym zainstalowaniu tzw. „tylnych drzwi" (ang. *backdoor*), które umozliwia w przyszlosci bezproblemowy powrót do systemu, tzn. bez pozostawiania sladów w postaci np. logów systemowych, z ominieciem systemów hasel itd. Jest to bardzo niebezpieczne dzialanie, gdyz pozwala wlamywaczowi przez dlugi czas po wlamaniu pozostac w ukryciu i sledzic praktycznie wszystkie aspekty naszej dzialalnosci np. przez podsluch naszej sieci.

Ponadto rozróżniamy nastepujace rodzaje ataków na bezpieczenstwo [4]:

?? przerwanie (*interruption*), czyli zniszczenie czesci systemu lub jej unieruchomienie, np. przeciecie linii lacznosci;

?? przechwycenie (*interception*), czyli uzyskanie dostepu do zasobów systemu przez czynnik postronny (osobe, komputer), np. podsluch;

?? modyfikacja (*modification*), tj. zmiana zasobów przez osobe nieupowazniona, np. modyfikacja programu lub komunikatów sieciowych;

?? podrobienie (*fabrication*), czyli atak na autentycznosc, np. dodanie falszywych danych do pliku.

2.3 Techniki wlaman

Techniki wlaman sa bardzo rózne od bardzo prostych do bardzo wyrafinowanych. Mozna je ogólnie podzielic na:

?? ataki pasywne (ang. *passive attack*), ataki na bezpieczenstwo, polegajace na podsluchiwaniu lub sledzeniu przesylania danych w celu odkrycia tresci komunikatu lub wykonania analizy ruchu (*traffic analysis*) danych w sieci. Poniewaz ataki pasywne nie zmieniaja danych, sa trudne do wykrycia.

?? ataki aktywne (ang. *active attack*), ataki na bezpieczenstwo, polegajace na wykonywaniu zmian w strumieniu danych lub tworzeniu danych falszywych. Rozróżnia sie cztery rodzaje ataku aktywnego: maskarade, atak przez ponawianie, modyfikowanie komunikatów oraz blokowanie dzialania. Modyfikowanie komunikatów moze równiez oznaczac ich opóznianie. Blokowanie dzialania ma utrudnic normalna prace systemu, np. niedocieranie informacji do miejsca, w którym sa sprawdzane, dezintegracje sieci lub jej przeladowanie [4].

2.3.1 Socjotechnika

Paradoksalnie, najprostsze metody (niekiedy bardzo skuteczne – jednak jest to sprawa indywidualna) nie wymagaja wielkiej wiedzy na temat luk w zabezpieczeniach systemów do których wlamywacz chce uzyskac dostep. Zaliczaja sie do tej grupy metody socjotechniczne (ang. *social engeneering*) polegajace na wyludzaniu hasel dostepu do systemu wprost od jego nieswiadomych uzytkowników. Najprosciej im zapobiegac stosujac polityke bezpieczenstwa oraz szkolac uzytkowników sieci [5].

2.3.2 Ataki slownikowe i „brute force"

Innym przykladem moga byc próby zgadywania prostych hasel stosowanych czesto przez nierozwaznych administratorów na standardowych kontach takich jak „ppp" czy „test" lub tzw. ataki slownikowe polegajace na sprawdzaniu kolejno hasel z wczesniej przygotowanej listy (slownik) oraz „*brute force cracking*" polegajacy na testowaniu wszystkich mozliwych kombinacji dostepnych znaków alfanumerycznych. Zabezpieczeniem przeciwko takim atakom jest stosowanie skomplikowanych hasel, skladajacych sie z wielu malych i duzych liter oraz cyfr.

2.3.3 Ataki „buffer overflow"

Duzo ciekawsze w dzialaniu i dajace spore mozliwosci lecz wymagajace od wlamywacza doglebnej wiedzy nie tylko z dziedziny programowania niskopoziomowego sa techniki tzw. ataków „buffer overflow" polegajace na stosowaniu tzw. exploitów (a konkretnie remote exploit czyli wersja dzialajaca zdalnie i szczególnie z tego wzgledu niebezpieczna) czyli specjalnie do tego celu napisanych programów, które wykorzystuja bledy w popularnych uslugach takich jak np. serwer IMAP i wiele innych. W skrócie pozwalaja one wykonac dowolny kod na odleglej maszynie Linuxowej poprzez wyslanie do niej odpowiednio spreparowanych danych, które uszkadzaja stos blednie napisanego programu serwera, czesto dzialajacego w trybie uprzywilejowanym (z ustawionym bitem **SUID**) [3, 5].

3. OCHRONA PRZED WLAMANIAMI

3.1 Polityka bezpieczenstwa

Polityka bezpieczenstwa do dokument opisujacy w jasny i czytelny sposób zasady bezpieczenstwa stosowane w organizacji. Powinna byc latwa do przestrzegania, chronic dane i prywatnosc uzytkowników. Czesto stosowana zasada polityki bezpieczenstwa jest zasada mówiaca, ze **to co nie jest dozwolone jest domyslnie zabronione**. Polityka bezpieczenstwa okresla wiele szczególowych zasad np.:

- ?? kto ma miec dostep do systemu i na jakich zasadach
- ?? jakie ma miec uprawnienia w systemie
- ?? czy moze uzywac cudzego konta lub uzyczac wlasnego konta innym
- ?? jaka jest odpowiedzialnosc za nieprzestrzeganie polityki
- ?? jakie programy mozna uzywac a jakich nie
- ?? co i komu mozna instalowac w systemie
- ?? kto moze dokonywac danych czynnosci np. instalacja oprogramowania, backup
- ?? itd...

Przestrzeganie polityki bezpieczenstwa powinno dotyczyc wszystkich uzytkowników systemu niezaleznie od rangi czy pozycji bo tylko w taki sposób spelni ona swoja role.

3.2 Firewall i testowanie zabezpieczen

3.2.1 Netfilter

Do podstawowych i zarazem standardowych narzedzi jakie oferuje Linux w zakresie ochrony systemu przed atakami z sieci jest wbudowany firewall. W obecnej wersji jadra linuxowego (2.4) standardowo dostepny jest **Netfilter** – narzedzie znane bardziej pod nazwa **iptables**. W odróznieniu od ipchains znanego z wersji 2.2 jadra linuxowego, Netfilter jest filtrem typu „*stateful – inspection*" [10].

Filtr stateful–inspection stoi o stopien wyzej od tradycyjnych zapór i skutecznie eliminuje ich niedogodnosci. Podstawa jego dzialania jest biezace sledzenie i analiza przechodzacych przez dany wezel, polaczen, co pozwala na znacznie skuteczniejsze kontrolowanie ich legalnosci. Filtr caly czas przechowuje w pamieci informacje na temat aktualnego stanu kazdego polaczenia, wie przy tym jakie kolejne stany sa dozwolone z punktu widzenia protokolu (IP/TCP/UDP), jak i polityki bezpieczenstwa.

Filtry tego typu pozwalaja na okreslenie mozliwosci dokonania danego polaczenia bez koniecznosci operowania poszczególnymi stanami protokolu TCP. Do administratora nalezy tylko okreslenie kierunku oraz polityki wzgledem rozpoczecia danego polaczenia, a filtr automatycznie weryfikuje kolejne etapy jego nawiazywania i pózniejszy przebieg. Ta ostatnia cecha pozwala równiez na odrzucanie pakietów, które do danej sesji nie naleza, co w praktyce przeklada sie na skuteczne blokowanie prób skanowania portów lub wprowadzania sfalszowanych pakietów (spoofing).

Przykladowo, klasyczny filtr pakietowy dla przepuszczenia pelnego polaczenia TCP do danego serwera potrzebowal co najmniej dwóch regul: wpuszczania pakietów do danego adresu i ich wypuszczania na zewnatrz. Rozbudowywanie tej polityki na przyklad o kierunek dozwolonych polaczen (czyli z której strony mozna je zaczynac) wymagalo dalszego rozbudowania listy, na przyklad o okreslenie ze rozpoczynajace polaczenie pakiety z flaga SYN sa wpuszczane tylko w danym kierunku. Dla filtra stateful–inspection w tym wypadku wystarczajaca jest wylacznie jedna regula, a mianowicie ze pakiety z flaga SYN sa wpuszczane do serwera na danym porcie. Pakiety bedace czescia polaczenia idace w obu kierunkach beda przepuszczane automatycznie. Równoczesnie jednak analogiczne, ale nie bedace czescia dozwolonego polaczenia pakiety zostana zablokowane [11].

3.2.2 Testowanie zabezpieczen – Nmap

Cennymi narzedziami sluzacymi do testowania bezpieczenstwa sa róznego rodzaju skanery. Odpowiednio uzyte pozwalaja na znalezienie wszelkiego rodzaju luk i niedociagniec w konfiguracji sieci. Skanowanie pozwala okreslic, czy dany host jest aktywny, dowiedziec sie o rodzaju dostepnych uslug i sprawdzic otwarte porty. Poza tym skanowanie pozwala poznac topologie sieci, oraz jej slabe punkty.

Jednym z najbardziej rozbudowanych i zaawansowanych skanerów jest Nmap [9]. W zakresie jego mozliwosci sa nastepujace techniki skanowania portów:

?? SYN/ACK

?? FIN

?? XMAS

?? NULL

Powyzsze techniki naleza do tzw. technik *stealth* czyli ukrytych. Z naszego punktu widzenia skanowanie portów jest szczególnie interesujace, gdyz pozwala testowac reguly firewalla i wynajdywac porty, które sa otwarte i stanowia potencjalne zagrozenie ze wzgledu na mozliwosc ataków exploitami.

3.3 Wykrywanie wlaman

Uzytkownicy a szczególnie administratorzy Linuxa spotykaja sie w swojej pracy z problemem wlasciwego zabezpieczenia swojego systemu, aby byl odporny na ataki czy próby dostania sie do niego z zewnatrz. Komercyjne systemy wykrywania intruzów sa zazwyczaj bardzo kosztowne, lecz istnieja równie dobre programy rozprowadzane na licencji GNU GPL [1]. Ogólnie, systemy wykrywania intruzów mozna podzielic na dwie grupy:

?? Systemy wykrywania ataków z zewnatrz (np. z Internetu)

?? Systemy wykrywania ataków z wewnatrz (z systemu operacyjnego)

3.3.1 Snort

Snort jest darmowym **systemem wykrywania intruzów z zewnatrz**, którego idea jest analiza zawartosci i typu przesylanych przez siec pakietów oraz ewentualne reagowanie na wykryte w ten sposób ataki. Korzysta on ze stale uaktualnianej i publicznie dostepnej bazy danych o atakach [7].

Bardzo wartosciowa cecha takiego rozwiazania jest rozbudowana mozliwosc konfiguracji. Dzieki prostemu jezykowi pisania regul oraz definiowania rodzajów i trybów postepowania w razie ataku, mozemy w latwy sposób dostosowac program do róznych typów aplikacji i architektury sieci. Snort ma równoczesnie szerokie mozliwosci okreslania akcji

podejmowanych przy atakach. Wspólpracuje on z róznymi bazami danych oraz na biezaco potrafi informowac wskazane przez nas konsole administracyjne (np. korzystajac z gniazd Unixa lub z Samby (Winpopup)) o wystapieniu niepokojacego zdarzenia.

Snort wykrywa ukryte skanowanie portów i zapisuje kazda próbe skanowania do pliku. Detekcja ta polega na monitorowaniu i zliczaniu pakietów, które przychodza na rózne porty maszyny w krótkich przedzialach czasu. Program moze pracowac zarówno w trybie wykrywania intruzów jak i w trybie nasluchu pakietów (ang. *sniffing*).

Pisanie dobrych regul wykrywania intruzów wymaga zaawansowanej znajomosci struktury budowy pakietów, formatu oraz zawartosci ich naglówków i przesylanych danych, jak równiez szerokiej wiedzy zwiazanej z uzywanymi aplikacjami, takiej jak np. wielkosc bufora, specyficzne wywolania danych procedur itp. Do wielu zastosowan przydaje sie znajomosc skladni jaka oferuje Snort. Za ich pomoca mozna bowiem sledzic dowolne dzialania uzytkowników i potencjalnych intruzów [2].

3.3.2 LIDS

LIDS (Linux Intrusion Detection System) jest systemem wykrywania intruzów glownie z wnetrza systemu operacyjnego [6]. W przeciwienstwie do Snorta, skupia sie bardziej na ochronie plików oraz pewnych operacji wykonywanych przez uruchomione procesy. Dzieki temu, ze jest wkompilowany w jadro systemu, ogranicza takze prawa root'a, wiec na przyklad podmienianie plików konfiguracyjnych czy binarnych (np. demonów) zwiazane jest z koniecznoscia rekompilacji i instalacji nowego jadra (na co oczywiscie dobrze skonfigurowany LIDS nie pozwoli). Trzeba pamietac, ze spowodowanie wykonania nieprawidlowej (lecz udanej) operacji przez jakikolwiek proces (np. demona SMTP) uruchomiony na prawach root'a, moze dac intruzowi nieograniczony dostep do systemu. LIDS kontroluje wszystkie próby korzystania z plików i katalogów, tak aby zapobiec wykonywaniu niedozwolonych operacji [2].

3.3.3 Podsumowanie

Powyzej przedstawione narzedzia stanowia duza pomoc dla administratora systemu dbajacego o bezpieczenstwo wlasnego serwera. Niestety nie stanowia panaceum na wszelkie troski administratora, gdyz bardzo zdolny haker potrafi równiez takie zabezpieczenia ominac. Dlatego

zadne narzedzie nie zwalnia osoby administrujacej serwerem do ciaglej czujnosci i opracowywania swoich oryginalnych i nietypowych metod zabezpieczania systemu. Stosujac powyzsze narzedzia nalezy pamietac o tym, aby kompilowac je na bezpiecznej maszynie. Kompilacja w srodowisku przejetym przez wlamywacza moze spowodowac, ze otrzymane binaria beda zmodyfikowane w taki sposób, ze wynik ich dzialania nie bedzie wiarygodny.

Koniecznie nalezy pamietac o bezpieczenstwie binarów programów uzywanych do testowania i nadzorowania bezpieczenstwa serwera. Chronic nalezy takze pliki konfiguracyjne oraz bazy danych zawierajace sygnatury waznych plików systemowych. Watro pamietac o utworzeniu kopii zapasowych najwazniejszych plików konfiguracyjnych serwera, aby móc je szybko odtworzyc w przypadku naruszenia ich zawartosci przez wlamywacza.

LITERATURA:

[1] M. Olejniczak „Narzedzia do detekcji wlaman", Linux+ nr 1/2002

[2] G. Landecki „Systemy wykrywania intruzów", Linux+ nr 1/2002

[3] A. Dudek „Nie tylko wirusy", Wydawnictwo Helion, 1998 r.

[4] Z. Ploski "Slownik Encyklopedyczny - Informatyka", Wydawnictwo Europa, 1999r.

[5] M. Hajder, H. Loutskii, W. Streciwilk „Informatyka – wirtualna podróz w swiat systemów i sieci komputerowych", Wydawnictwo WSIiZ, Rzeszów 2002 r.

[6] Projekt LIDS: **http://www.lids.org/**

[7] Projekt SNORT: **http://www.snort.org**

[8] Linux Security HOWTO: **http://www.linux.org/docs/ldp/howto/Security-HOWTO/**

[9] Nmap: **http://www.insecure.org/nmap/**

[10] Netfilter: **http://www.netfilter.org/**

[11] P. Krawczyk „Filtrowanie stateful-inspection w Linuxie i BSD", 2001 r.