Systémy pro detekci a prevenci průniků

Lukáš Kokrment, Jan Pavlovič
{xkokrmen, xpavlov}@fi.muni.cz

Fakulta informatiky
Masarykova univerzita
Botanická 68a, 602 00 Brno
Česká republika

Abstrakt

S ohledem na současné masové rozšíření Internetu a ostatních informačních technologií se do popředí zájmu profesionálů (administrátorů), manažerů i běžných uživatelů stále více dostávají otázky bezpečnosti počítačových systémů. Počet bezpečnostních incidentů (pokusů o průniky i úspěšných průniků) v současnosti prudce stoupá a proto je nutné přijímat aktivní a účinná opatření, která budou minimalizovat možnosti napadení počítačových systémů.

V posledních letech se velmi diskutovanou oblastí staly systémy, které jsou schopny analyzovat síťový provoz (obsah, chování …), podle těchto analýz rozpoznat, jestli je tento druh provozu legitimní, nebo jestli představuje potencionální hrozbu, a na základě tohoto zjištění např. upozornit administrátora. Hovoříme o tzv. systémech pro detekci průniků (*intrusion detection systems*). Nyní se tyto systémy často vyznačují možností automatického přijímaní různých aktivních opatření, které mají za cíl při podezření na nekalou činnost zamezit útoku (např. zrušení spojení, spolupráce s firewallem …) a hovoříme o systémech pro prevenci průniků (*intrusion prevention systems*).

1. Úvod

Počítačová bezpečnost se stala v poslední době jedním z nejvíce diskutovaných témat v oblasti informačních a telekomunikačních technologií (ICT). Téměř každá moderní firma se dnes již téměř neobejde bez fungující počítačové infrastruktury a připojení na Internet. Tento trend bude časem v souvislosti s přibližováním České republiky k západním zemím stále markantnější. Již dnes můžeme často slyšet názor, že kdo není na Internetu, jako by neexistoval.

Toto globální používání informačních technologií a zejména Internetu však s sebou přináší mnohé problémy. Původní Internet byl navržen především s ohledem na robustnost sítě, avšak pouze s minimálním důrazem na bezpečnost počítačových systémů, které by měla tato síť spojovat. U Internetu se předpokládalo, že bude spojovat spoustu „přátelských" systémů a uživatelů, kteří budou spolu komunikovat, sdílet informace a data, a využívat síť více méně k vědeckým účelům. Problémům bezpečnosti se při návrhu internetových protokolů věnoval málokdo.

Dnes můžeme pozorovat mnoho problémů, které vyplývají jednak z nedostatků a chyb aplikací, které mají pracovat v síťovém prostředí (útoky na dostupnost služeb, získání neoprávněného přístupu k systému, …) a jednak z nedostatků v samotném návrhu protokolů, které tyto aplikace používají a na kterých je existence internetu založena (falšování adres, …).

2. Typy útoků

Jak již bylo zmíněno v úvodní kapitole, můžeme obecně rozeznávat podle cíle dva základní typy útoků. Útoky na funkčnost sítě jako takové (*network level attacks*) [1] a útoky na zranitelná místa v aplikacích (*application level attacks*) [1].

Pokud se podíváme na OSI [2] model síťové architektury, který se skládá ze sedmi vrstev, můžeme říci, že útoky na síťovou architekturu se týkají páté vrstvy (*session layer*) a nižších. Útoky na aplikace se týkají prezentační nebo aplikační vrstvy (*presentation, application layer*).

Jako typické útoky na síť můžeme uvést následující:

- ARP flooding [3] – v dnešní době dominují v praxi téměř výhradně přepínané sítě (pokud neberme v úvahu stále více se prosazující bezdrátové technologie). Přepínané sítě mají na rozdíl od klasického sdíleného média s ohledem na bezpečnost tu výhodu, že aktivní prvek (*switch*) se snaží doručit daný síťový provoz pouze té cílové stanici, které je opravdu určen. Tento síťový provoz tedy nevidí a nemohou odposlechnout ostatní stanice na síti. Aby tato komunikace 1:1 byla možná, musí znát síťový přepínač cestu k cílové stanici, musí vědět, kam má data poslat. Ten si tedy udržuje paměť hardwarových adres, které náleží jednotlivým zařízením. Pokud přepínač neví, kam má poslat data, pošle je všem a čeká, která stanice odpoví. Potencionální útočník se tedy může pokoušet zahltit paměť přepínače tak, aby ten nebyl schopen efektivně využívat svou paměť a musel posílat veškerá data (nebo jejich naprostou většinu) všem stanicím a tedy i útočníkovi.
- IP spoofing [3] – falšování IP adresy může umožnit útočníkovi uklidit stopy po svém působení a ztížit jeho odhalení. IP spoofing může útočník využít např. k pokusu o útok na dostupnost služby (*denial of service*) [3], kdy se pokusíme zahltit cílový systém množstvím paketů s falešným odesílatelem.
- Skryté kanály (*covert channels*) [3] – skryté kanály samy o sobě nepředstavují útok. Skryté kanály představují použití skrytých cest pro přenos dat či pro komunikaci. Takovým skrytým kanálem může být např. použití některých polí ICMP datagramu. Skryté kanály mohou mít mnoho případů legitimního použití, ale mhou sloužit např. také útočníkovi ke komunikaci s napadeným systémem, která je jen velmi těžce odhalitelná.
- a mnoho dalších …

Mezi typické techniky nebo představitele útoků na aplikační úrovni pak patří např.:

- Přetečení bufferu *(buffer overflow)* [3]– tento typ zranitelnosti je v současnost snad nejčastější a je typický pro špatně naprogramované aplikace. Typicky vzniká, pokud si v programu vyhradíme určité místo, do kterého chceme zapsat data, avšak útočník programu podstrčí množství dat, které program neočekává. Pokud si program data nezkontroluje, dojde k přepsání paměti programu, které může mít neočekávané důsledky. Útočník se většinou snaží pomocí přepsání paměti pozměnit běh programu, případně provést vlastí kód. Následující kousek kódu ukazuje příklad přetečení bufferu:

```
void main(int argc, char** argv)
{
  char buffer[5];
  strcpy(buffer, “Tento text zpusobi preteceni”);
};
```

- Útoky pomocí formátovacích řetězců – tento typ útoků umožňují opět zejména špatně naprogramované aplikace. Vzniká typicky v programech napsaných v jazyce C, pokud se programátor pokusí načíst vstup přímo do proměnné bez použití formátovacích řetězců.
- Viry a červy – viry jsou programy nebo části programů, které se mají šířit a typicky provádět na počítači oběti nějakou činnost, kterou si uživatel nepřeje, neví o ní (nemusí přímo působit škody). Zatímco počítačový virus se typicky šíří za pomoci

uživatele (spuštěním napadeného souboru, …), červ se šíří ze systému na systém typicky bez pomoci uživatele, když zneužívá nějakou chybu v aplikaci (přetečení bufferu u síťového démona, …).

- a mnoho dalších …

3. Systémy pro detekci průniků

Pod pojmem systém pro detekci průniků (*intrusion detection system*, IDS) [4,8] si můžeme představit soubor nástrojů a metod, jejichž cílem je odhalit aktivitu, jejímž cílem je nebo může být průnik do systému, neoprávněný přístup k informacím, pokus o útok na dostupnost systému nebo služby apod. a upozornit na tuto nekalou či neoprávněnou aktivitu.

Dalším stupněm ve vývoji, který se dnes stal již téměř „zaklínadlem" firem podnikajících v oblasti počítačové bezpečnosti je systém na prevenci průniků (*intrusion prevention system*, IPS) [4]. Zatímco systémy pro detekci průniků mají sledovat podezřelou aktivitu a tuto aktivitu upozornit (většinou administrátora), systémy pro prevenci průniků jejich funkcionalitu rozšiřují o možnost přijímat aktivní opatření proti potencionálním útočníkům. Mezi tato aktivní opatření může patřit např. spolupráce s firewallem, kdy systém může při pokusu o průnik automaticky začít zahazovat veškerý síťový provoz pocházející od útočníka. Dalším příkladem aktivního opatření může být reset síťového spojení (zaslání paketu s RST příznakem) nebo pokud je podezřelá aktivita spojena s nějakým uživatelem, může systém pro prevenci průniků odhlásit daného uživatele a zamezit mu v dalším přihlášení, případně zabít podezřelý proces (procesy).

Přestože se v poslední době tyto systémy staly velmi populárním nástrojem, musíme mít na paměti, že nepředstavují všelék na problémy počítačové bezpečnosti. Systémy pro detekci či prevenci průniků v žádném případě nedokáží nahradit ostatní bezpečnostní prvky, jako jsou firewally, antivirové programy, včasnou aktualizaci softwaru aj. Jejich nasazení by mělo být vždy součástí komplexní bezpečnostní politiky organizace.

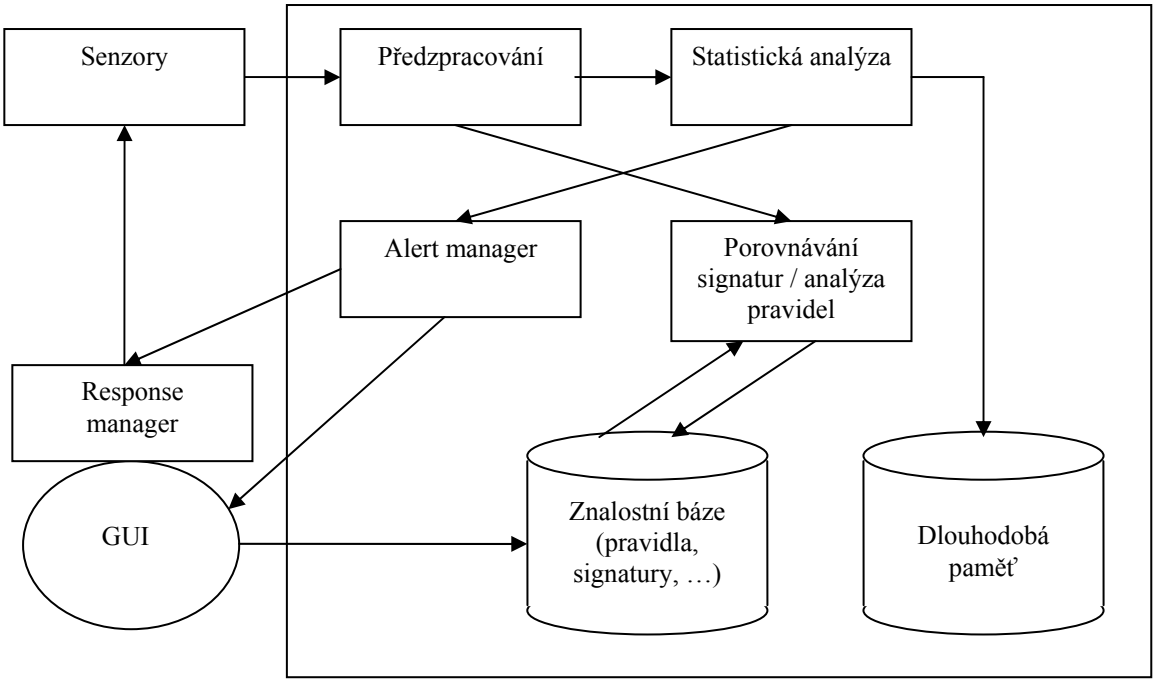
3.1 Architektura

Typicky můžeme systémy pro detekci průniků rozdělit podle jejich architektury a způsobu nasazení do tří základních kategorií.

Systémy běžící na jednom konkrétním počítači se označují anglicky jako *host based* a slouží typicky k monitorování důležitých pracovních stanic a serverů. *Host based* systémy typicky fungují na základě kontroly logů daného systému, analýzy síťového provozu týkající se konkrétního systému, kontroly souborů a souborového systému apod. Za *host based* systémy na detekci průniků můžeme považovat např. i populární systémy kontrolující integritu souborového systému (Tripwire, AIDE, …) [4].

Síťové systémy pro kontrolu průniků (*network based*) oproti předchozím pracují v rámci celé sítě či podsítě a jejich činnost je založena na analýze síťového provozu, který se vyskytuje na daném segmentu sítě. Zatímco na klasických ethernetovských sítích jsme mohli zachycovat veškerý síťový provoz jednoduše přepnutím síťové karty do tzv. promiskuitního módu, na přepínaných sítích, které dnes převažují, musíme řešit problém, že síťové přepínače obvykle posílají daným stanicím pouze síťový provoz, jehož jsou adresátem. Naštěstí většina přepínačů umožňuje použití technik jako *spanning port*, které nám umožňují duplikovat síťový veškerý síťový provoz na dané rozhraní.

Třetí možností je použití hybridního přístupu, kde systém tvoří několik spolupracujících komponent pracujících na několika počítačích, které jsou schopny monitorovat jak síťový provoz, tak jednotlivé počítače a komunikovat spolu buď v režimu peer-to-peer, kdy jsou všechny komponenty na stejné úrovni, nebo v režimu klient-server, kdy existuje jedna dedikovaná komponenta, která řídí a spravuje ostatní.



Obrázek č. 1: Obecná struktura IDS (Endorf 2004)

Na obrázku číslo 1 můžeme vidět znázorněnu zjednodušenou architekturu typického systému pro detekci průniků. Základní a možná i nejjednodušší komponentou systému jsou senzory. Ty slouží podle druhu IDS buď k zachytávání síťového provozu, k získávání informací ze systémových logů apod. V síťovém prostředí může systém pro detekci průniků zpracovávat informace pocházející z mnoha různých senzorů umístěných např. na různých podsítích. V případě síťového nasazení se jako velmi vhodné ukazuje umístění senzorů hned za firewall či routek (avšak na fyzicky jinou stanici), kde mají přístup k veškerému provozu, směřujícím z Internetu do vnitřní sítě. Nesmíme však zapomínat ani na provoz na vnitřní síti (většina útoků na počítačové systémy dnes pochází z vnitřní sítě).

Ve fázi předzpracování (*preprocessing*) se provádí s daty získanými ze senzorů operace jako je např. skládání jednotlivých fragmentů síťového provozu, dekódování jednotlivých protokolů, oddělení hlaviček paketů a jejich datových částí (pokud chceme analyzovat pouze na základě jednoho z výše uvedených), ošetření šifrovaných dat (můžeme je chtít dešifrovat nebo se je rozhodneme vůbec neanalyzovat) atd.

Předzpracovaná data jsou pak předána na vlastní analýzu, při které se systém snaží odhalit podezřelou aktivitu. Analýza se provádí na základě informací, které má systém uloženy obvykle ve formě signatur nebo pravidel ve své znalostní bázi. Metodami této analýzy se budeme zabývat v další kapitole.

Údaje o analyzovaných datech, záznamy o podezřelé aktivitě a další údaje jsou obvykle zaznamenávány v nějaké dlouhodobé paměti, aby byly k dispozici pro další použití či k vytváření přehledů.

Celý systém pak bývá většinou řízen administrátorem pomocí manažerských komponent, díky kterým je schopný reagovat na upozornění a výsledky analýzy a nastavovat parametry sytému. K dispozici může být i grafické uživatelské rozhraní (GUI).

3.2 Metody analýzy dat

Systémy pro detekci / prevenci průniků používají pro analýzu dat zachycených senzory a pro rozhodnutí, jestli je daná aktivita nebo provoz potenciálně nebezpečná či nikoliv, většinou některou následujících technik:

Porovnávání signatur (*signature matching*) [5] je nejjednodušší, ale patrně také nejméně efektivní metodou analýzy dat. Pod pojmem signatura můžeme chápat posloupnost znaků, která je charakteristická pro určitý typ činnosti nebo pro určitý druh útoku. Může se jednat např. o posloupnost znaků, která se vyskytuje v těle určitého počítačového viru, charakteristickou sekvenci, která může být součástí shell kódů, pomocí kterých se snaží útočník zneužít přetečení bufferu, atd. Soubor těchto signatur bývá umístěn ve znalostní bázi systému, který pak provádí porovnání zachycených dat s touto znalostní bází. Problémem bývá, jak přesně provádět porovnání se známými signaturami. Přesné porovnávání (*exact matching*) [5] bývá často neefektivní. V praxi např. existuje většinou mnoho variant počítačových virů, které se navzájem liší pouze v drobnostech, některé tyto viry pak samy záměrně mutují a mění se. Bývá tedy obtížné vybrat jednu posloupnost znaků, která je charakteristická ve všech případech. Při porovnávání signatur se tedy používají zpravidla různé techniky pro částečnou shodu (*partial matching*) [5] a ohodnocení „síly této shody".

Porovnávání na základě pravidel (*rule based matching*) [5] představují po signaturách kvalitativně další stupeň analýzy dat získaných ze senzorů. Analýza na základě pravidel kombinuje různé faktory a na jejich základě rozduje, jestli je aktivita legitimní či jestli představuje nebezpečí. Systém založený na analýze pravidel může např. zůstat potichu, pokud se cizí počítač pokusí zaslat data na určité porty, může ale zalarmovat správce, pokud se tato data pokusí poslat poté, co provedl skenování portů systému. Signatury známých útoků mohou být někdy součástí pravidel. Obecně můžeme říci, že analýza pomocí pravidel je efektivnější než analýza založená čistě na signaturách a generuje menší množství falešných poplachů.

Třetí a pro naše účely poslední možností, kterou mohou systémy na detekci či prevenci průniků používat je analýza na základě profilů (*profile based matching*) [5]. Analýza na základě profilů je spíše doménou *host based* systémů, zatím se však nejedná o příliš rozšířenou techniku. Systém si nejprve uloží informace o typickém chování objektu např. uživatele a hledá změny v typickém chování objektu a aktuálním chování objektu. Příkladem může být sledování chování uživatelů. Pokud se např. zaměstnanec typicky přihlašuje na server ze své stanice v průběhu pracovní doby a systém zaznamená najednou jeho přihlášení v nočních hodinách z počítače umístěného někde v Asii, může se jednat o průnik do systému. Výhodou tohoto přístupu může být fakt, že na rozdíl např. od porovnávání na základě signatur definujeme spíše to, co je dovoleno a hledáme to, explicitně povoleno není, než naopak.

3.3 Výhody a nevýhody nasazení IDS / IPS

V předchozích kapitolách jsme se seznámili s obecnými principy fungování systémů na detekci / prevenci průniků. Nyní zmíníme více explicitně některé výhody a nevýhody, které plynou z jejich nasazení v praxi:

- Lepší zabezpečení počítačové sítě – pokud jsou systémy tohoto typu nasazeny správně, jsou dobře nakonfigurovány a administrátor jim věnuje patřičnou pozornost, mohou výrazně zvýšit úroveň zabezpečení sítě.
- Systémy pro detekci / prevenci průniků umožňují nalézat efektivněji zranitelná místa v počítačové síti.
- Systémy pro prevenci průniků nám poskytují možnost rychlé reakce na bezpečnostní incidenty a předcházení jim.
- Výhodou může být také možnost automatizace odpovědí na bezpečnostní incidenty.
- Nezanedbatelnou výhodou mohou být také možnosti logování, analýzy těchto logů a tvorby reportů.

Nasazení IDS / IPS však nepředstavuje všelék na problémy počítačové bezpečnosti. Jak již bylo zmíněno dříve, toto nasazení by mělo být pouze součástí celého systému bezpečnostních opatření, nikoliv jejich náhradou. A protože nic není zadarmo, můžeme zmínit i některé nevýhody či problémy:

- Netriviální konfigurace IDS / IPS – správné nastavení systémů pro detekci průniků je často velice pracné, jejich vyladění vyžaduje čas a zkušenosti.
- Pokud systém není dobře nakonfigurován, může dávat vysoký počet falešných poplachů (což po čase pochopitelně oslabí pozornost obsluhy) nebo, co je možná ještě horší, bude ignorovat některé hrozby.
- Problémy s šifrovanými daty – pokud nejsme schopni data rozšifrovat, nejsme je schopni ani analyzovat, nemusíme být tedy schopni detekovat útoky na určité aplikace.
- Problémy s vysokorychlostními sítěmi – některé systémy nejsou schopny analyzovat v reálném čase provoz na vysokorychlostních sítích.

4. Praktické použití a reálné systémy

Zatímco v předchozím textu jsme se zabývali spíše obecnými principy fungování IDS / IPS, v této části textu bychom rádi popsali vlastnosti několika konkrétních systémů pro detekci průniků.

4.1 Snort

Snort [6,7] je volně šiřitelný open source projekt, který je momentálně patrně nejoblíbenějším a nejvíce používaným síťovým systémem pro detekci průniků. Za jeho současnou popularitou stojí nejen snadná dostupnost, ale také jeho kvalita, rychlost aktualizací a vydávání nových pravidel a množství dalších nástrojů a modulů, které jsou vyvíjeny open source komunitou. Samotný Snort pracuje zejména nad protokolem TCP/IP, ale pomocí modulů jej můžeme rozšířit i o možnosti práce s dalšími síťovými protokoly, např. IPX. Systém je dostupný zejména pro UNIXovské platformy (GNU/Linux, *BSD, …), ale existuje i verze pro Windows.

Systém může běžet v několika režimech, pole parametrů, které mu předáme při startu. Snort může tedy fungovat jako sniffer nebo logger, kdy pouze poslouchá a zachytává síťový provoz. Tohoto režimu můžeme využít např. pro kontrolu konektivity, sledování datových toků na síti apod. Systém je schopný ukládat informace o zachyceném síťovém provozu jak do standardních textových souborů, tak do databází. Pro nás mnohem zajímavější je drhá možnost využití a to jako efektivního zejména síťového IDS.

Snort patří mezi systémy pro detekci průniků založené na pravidlech. Jako základ pravidla slouží signatury, k jejich vyhodnocování však Snort používá ještě mnoho dalších kontextových informací. Jednoduché pravidlo, které identifikuje ICMP ping s parametrem TTL s hodnotou 100 (Rehman 2003):

```
alert icmp any any -> any any (msg: "Ping with TTL=100"; \
      ttl:100;)
```

Každé pravidlo se obecně skládá z hlavičky a těla pravidla. Hlavička pravidla se skládá z popisu akce, která se má provést, protokolu, kterého se pravidlo týká a údajů o zdroji a cíli provozu. Tělo pravidla pak obsahuje další informace, které mohou sloužit k lepšímu vyladění pravidla, jeho popisu, …

Literatura

[1] Paul Serrano. Enterprise Security Solutions
http://www.networkmagazineindia.com/200402/inperson01.shtml

[2] International Organization for Standardization. Open System Interconnection
http://www.iso.org/

[3] Endorf C., Schultz E., Melander J., 2004. Intrusion Detection & Prevention. McGraw & Hill.

[4] Rehman R. U., 2003. Intrusion Detection Systems with Snort – Advanced IDS Techniques Using Snort, Apache, MySQL, PHP and ACID. Prentice Hall.

[5] Toxen B., 2002. Real World Linux Security: Intrusion Prevention, Detection and Recovery. Prentice Hall.

[6] Beale J. et al., 2004. Snort – Intrusion Detection. Syngress.

[7] Snort FAQ. Dostupné na adrese http://www.snort.org/docs/faq.html

[8] Intrusion Detection FAQ. http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm