Autentizace a identifikace

MARTIN IGNJATOVIĆ

VYLEPŠETE SI BEZPEČNOST VLASTNÍMI SILAMI

Každý, kdo používá počítač připojený k síti, se téměř neustále hlásí k různým síťovým službám a využívá cizí zdroje, jinak by ostatně ani počítačové sítě neměly smysl. Většina služeb v dnešní době využívá jistou formu autentizace uživatele, a to různými metodami. Málokdo se však zamýšlí nad tím, jak taková autentizace vlastně funguje a jak je bezpečná. Toto malé zamyšlení je cílem dnešního článku.

Šifrované versus nešifrované přenosy

Mnoho síťových autentizačních metod používá různé kryptografické algoritmy k šifrování vzájemné komunikace. Bohužel jsou i takové, které tak nečiní. Nyní se podíváme na ty protokoly, které k autentizaci šifrování nepoužívají.

Každý z vás již jistě dobře ví, že protokoly jako je FTP, POP, Telnet nebo HTTP šifrování téměř nevyužívají. Většina uživatelů ale nad tímto faktem mávne rukou a řekne si, že s tím nemůže osobně nic udělat, a dále se problémem netrápí. Situace však není tak růžová a jednoduchá, jak by se na první pohled mohlo zdát. Zejména v komerční sféře může mít takové ignorování faktů kritické následky, nemluvě o narušení soukromí, které není jistě příjemné nikomu. Na následujících několika řádcích si ukážeme, jaké může mít používání nešifrovaných protokolů následky, a co vše lze s trochou šikovnosti zjistit. Nakonec se podíváme na některé bezpečné alternativy nešifrovaných služeb a povíme si, jak pomocí nich řešit stávající situaci.

IDENTIFIKACE SERVERU A SLUŽEB

První věcí, kterou je třeba zjistit, je, jakou vlastně verzi a typ služby provozujete nebo využíváte. Pokusíme se identifikovat různé služby a jak jistě uvidíte, pokud nejsou správně nastaveny, prozradí tyto služby o sobě velmi mnoho. Prvním krokem by měla být identifikace cílového systému, který službu nabízí. Může jít o poštovní server, FTP server nebo webový server. Jako příklad jsem vybral systémy, které nabízejí mnoho služeb (což mimochodem není příliš šťastné a bezpečné řešení). V praxi se můžete setkat (je to velmi pravděpodobné) s tím, že služby budou nainstalovány na různých systémech. Uvedené služby běžely pod těmito operačními systémy: Linux RedHat 7.3 a Windows 2000 Professional. Na systémech byly spuštěny tyto služby: SMTP server, FTP server a HTTP server. Na systému Linux ještě SSH server a na systému Windows telnet server. Nyní se podíváme, jak vypadal portscan každého systému a jaké služby identifikoval. Ke skenování portů byl použit nástroj nmap *(www.insecure.org/nmap)*, který platí za jeden z nejlepších produktů své třídy.

Jak vidíme na obrázcích na vedlejší straně, nabízí každý systém podobné služby. Nyní se pomocí nástroje telnet podíváme na to, jak se jednotlivé služby identifikují. Alternativně lze použít i nástroj netcat, který je součástí většiny linuxových distribucí a je k dispozici i pro operační systém Windows.

Z pohledu správce

Nyní uděláme malou odbočku a podíváme se na problém ze strany správce systému. Z výpisu je vidět, že se většina služeb identifikuje celkem obsáhle, což ale nemusí být vždy dobře. Pokud útočník zná druh a verzi dané služby, může zjistit, byla-li v té které konkrétní verzi dané služby objevena nějaká chyba, a případně takovou chybu zneužít. S tím souvisí sledování změn a pravidelná aktualizace softwaru. Jak můžete vidět z výpisů, testované systémy jsou zprovozněny v základní konfiguraci a ke všemu ještě neaktualizované, což by se u systémů připojených k síti (a nejen u nich) stávat nemělo.

Nyní si tedy povíme, jak znesnadnit útočníkovi identifikaci běžících služeb. První a základní věcí je vůbec nespouštět služby, které nepotřebujeme. Proč mít na své lokální stanici spuštěnu službu telnet, když nechceme, aby náš počítač využíval někdo jiný? Další věcí je oddělení služeb, které slouží uživatelům vnitřní sítě, a služeb, jež chceme nabídnout veřejnosti. Zde přicházejí ke slovu firewally a směrovače. Rovněž není od věci uvažovat o implementaci takzvané DMZ neboli demilitarizované zóny, do které jsou umístěny služby nabízené veřejnosti, jež chceme oddělit od naší vnitřní sítě. Do DMZ můžeme umístit poštovní servery, FTP server nebo webové servery. Konfigurace a strategie firewallu však není tématem tohoto článku, a tak se raději vrátíme k našemu tématu a povíme si, jak znesnadnit identifikaci běžících služeb.

IDENTIFIKACE SLUŽEB NA SYSTÉMU WINDOWS 2000

```
[root@ init.d]# telnet 10.33.4.25 21
Trying 10.33.4.25...
Connected to 10.33.4.25.
Escape character is '^]'.
220 w2k Microsoft FTP Service (Version 5.0)

[root@ init.d]# telnet 10.33.4.25 25
Trying 10.33.4.25...
Connected to 10.33.4.25.
Escape character is '^]'.
220 w2k Microsoft ESMTP MAIL Service,
Version: 5.0.2172.1 ready at Wed, 26 Feb 2003
08:59:48 +0100

Server: Microsoft-IIS/5.0
Date: Wed, 26 Feb 2003 08:02:19 GMT
Content-Length: 3221
Content-Type: text/html
```

IDENTIFIKACE SLUŽEB NA SYSTÉMU LINUX REHHAT 7.3

```
[root@ init.d]# telnet 10.33.4.131 21
Trying 10.33.4.131...
Connected to 10.33.4.131.
Escape character is '^]'.
220 machine.example.com FTP server
(Version wu-2.6.2-5) ready.

[root@ init.d]# telnet 10.33.4.131 25
Trying 10.33.4.131...
Connected to 10.33.4.131.
Escape character is '^]'.
220 machine.example.com ESMTP Sendmail
8.11.6/8.11.6; Wed, 26 Feb 2003 08:53:05 +0100

<ADDRESS>Apache/1.3.27 Server at
machine.example.com Port 80</ADDRESS>
```

Jako první si ukážeme, jak odstranit hlavičku poštovního serveru Sendmail. Tuto změnu můžete provést pomocí konfiguračních maker m4 nebo přímou editací souboru *sendmail.cf*, což je jednodušší, neboť jde o změnu jednoho řádku, a to konkrétně o změnu řádku obsahujícího direktivu SMTPGreetingMessage=j Sendmail $v/$Z; $b. Tuto direktivu můžete libovolně měnit, můžete se dokonce vydávat za jiný poštovní server. Možností je mnoho a rozhodnutí leží jen na vás. Pozor ale na situaci, kdy se o poštovní server stará více správců a nastavení se provádí pomocí maker m4. Potom je důležité, aby změna byla provedena i v makru, aby nedošlo k situaci, kdy druhý správce změní nějaké konfigurační direktivy, znovu vygeneruje a zavede konfigurační soubor, a všechny změny pak přijdou nazmar. Dalším velice často používaným poštovní serverem je Postfix. I u něj si ukážeme, jak změnit jeho hlavičku. Změnu provedete v konfiguračním souboru Postfixu main.cf, kde je třeba upravit direktivu *smtpd_banner*. Další službou, kde je žádoucí změnit hlavičku, je FTP. Povíme si, jak to udělat u nejrozšířenějšího FTP serveru wu-ftpd. V konfiguračním souboru ftpaccess editujte direktivy *greeting full, greeting brief, greeting terse, greeting test, banner a hostname.* Pokuste se s těmito direktivami experimentovat a zvolte tu, která vám bude nejvíce vyhovovat. U HTTP serveru, což je v unixovém světě nejčastěji server Apache a ve světě Microsoftu nejčastěji IIS, je situace poněkud obtížnější. Například u serveru Apache se při změně hlavičky nevyhneme editaci zdrojového kódu a následné kompilaci. Změnu hlavičky provedete v hlavičkovém souboru *httpd.h*, který je uložen v adresáři *apachesource/include,* kde *apachesource* je adresář, v němž máte uloženy zdrojové kódy serveru. V tomto souboru změňte hodnotu makra SERVER_BASEPRODUCT a SERVER_BASEVERSION. Verzi HTTP serveru zjistíte velice snadno zasláním následujícího požadavku na port, na kterém HTTP server běží: HEAD /HTTP/1.0. Nyní se vraťme do pozice uživatele, jenž se snaží identifikovat běžící služby a zjišťuje jejich vlastnosti.

Z pohledu uživatele

Dejme tomu, že se vám podařilo více či méně úspěšně identifikovat služby, které využíváte, a z dostupných informací, jimiž bývají nejčastěji domovské stránky jednotlivých produktů, jste zjistili, zda služba šifruje komunikaci a na jakém principu provádí autentizaci. Nyní se budeme zabývat postupem pro případ, kdy zjistíte, že služba šifrování nevyužívá. Abyste pochopili význam nebezpečí, které vám hrozí, uděláte nejlépe, když si sami zkusíte, čeho je potenciální útočník schopen. Nejjednodušším způsobem, jak obejít síťovou autentizaci, je použití snifferu a odchycení hesla právoplatného uživatele. To je mnohem snazší, než by se mohlo zdát. Jako sniffer je použit produkt ettercap *(sourceforge.net/ettercap),* který nabízí špičkový výkon, mnoho standardních funkcí, další užitečné plug-iny, a navíc využívá GTK interface pro uživatele, kteří mají rádi grafické prostředí. Pro konzolové uživatele je vedle klasického rozhraní přichystáno i rozhraní využívající ncurses. Program umí různé druhy sniffu, ať již jde o ARP sniffing užitečný zejména na přepínaných sítích, IP sniffing nebo MAC sniffing. V případě, že na vaší síti běží služba arpwatch, která sleduje změny na síti, a vy se pokusíte o ARP sniffing, bude arpwatch hlásit spoustu nestandardních situací, zejména flip-flop sniffovaného stroje. Proto raději na své aktivity předem upozorněte správce sítě.

Nyní již tedy k samotnému testování nešifrovaných služeb. Myslíte si, že je vaše pošta v bezpečí před cizími zraky? Zkuste zapnout sniffer, a jako source adresu vyberte poštovní server a jako destination vyberte svůj systém a zapněte sniffování (typ zvolte podle požadavků sítě). Nyní na svém stroji zkuste přijmout/odeslat poštu. Vidíte výsledky? To, co vidíte, může vidět kdokoliv na vaší síti, případně každý, kdo má do vaší sítě přístup. Si-

tuace však může být ještě horší. Podívejte se na obrázek vlevo dole. Zde jsme spuštěním příslušného plug-inu zaznamenali celou komunikaci mezi serverem a klientem, jinými slovy udělali jsme si kopii e-mailu. Co se stane v případě, kdy poštou posíláte citlivé informace, mající zásadní vliv na fungování vaší firmy?

Nyní se podíváme na protokol ftp. Opět si zapněte sniffer a pokuste se připojit k libovolnému ftp serveru. Vidíte vaše jméno a heslo? Pokud ano, může ho vidět i kdokoliv ve vaší síti, případně na cestě mezi vámi a serverem, ale tuto situaci již opravdu asi neovlivníte. Co se stane v případě, kdy stahujete citlivá data z firemního ftp serveru, o kterém si myslíte nebo o němž vám bylo tvrzeno, že je zabezpečen?

Dobře, řeknou si někteří lidé, a co když omezím přístup jen na povolené IP adresy? I proti tomuto lze vznést námitku, neboť IP adresa může být velice snadno zfalšována či podvržena. Nyní se ještě podíváme na HTTP komunikaci, abychom si ukázali, že ani tento druh přenosu není příliš bezpečný. V prostředí webu se používá mnoho autentizačních metod. Některé jsou založené na funkcích serveru, některé na skriptovacích jazycích, jež na serveru fungují. Velice často se však používá HTTP autentizace, nejčastěji pak typu basic. S tímto druhem autentizace se jistě každý již někdy setkal.

Bohužel i tento typ autentizace lze velice snadno obejít a překonat. Opět spustíme náš oblíbený ettercap a budeme chytat veškerý provoz mezi webovým serverem a klientem. V okně prohlížeče otevřeme stránku a pokusíme se vstoupit do nějaké autorizované oblasti, která je chráněna HTTP autentizací. Pokud nemáme takový komfortní nástroj jako je ettercap, jenž nám naservíruje všechna hesla jako na podnose, ale máme k dispozici „pouze“ starý dobrý sniffer, dostaneme výstup podobný tomuto:

GET /adresar1/adresar2 HTTP/1.0
Autorization: Basic *upravena podoba hesla*

Jelikož heslo není šifrované, ale pouze upravené pro přenos po síti, je získání jeho podoby otázkou několika sekund. Stačí nám k tomu například skriptovací jazyk PHP, Perl atd. Zkuste Perl a base_decode64 nebo její obdobu v PHP. Další perličkou (slovo nemá souvislost se skriptovacím jazykem Perl ☺) programu ettercap je plug-in, který umožňuje zachytávat všechny kopírované soubory přes protokol HTTP.

Existují také různé autentizační metody, jež využívají sessions, cookies nebo například hashů md5 algoritmu. Všechny uvedené metody jsou více či méně bezpečné, ale nemá cenu je zde podrobně rozebírat. Za malou chvilku se podíváme na metodu, která je při správné implementaci ze všech uvedených nejbezpečnější.

Alternativy

Jestli vás předchozí řádky dostaly do deprese a zoufáte kvůli tomu, že žádná data nejsou v bezpečí, tak buďte v klidu. Existují metody, jak svá data chránit a jak nahradit stávající nebezpečné služby jejich bezpečnější alternativou. Nyní si uvedeme krátký přehled alternativních služeb.

POP, IMAP

Většina poštovních klientů ani POP nebo IMAP serverů šifrování nepodporuje. Řešením jsou servery APOP a KPOP, které fungují na trochu odlišném principu než klasický POP server. Princip komunikace spočívá v tom, že server pošle klientovi požadavek, tento požadavek klient zpracuje tak, že pomocí hesla vygeneruje odpověď, kterou odešle zpět serveru. K přenosu hesla přes síť tak vůbec nedochází. Nasazení těchto serverů je bohužel velmi malé. Pokud však máte vliv a zájem na bezpečnosti dat, nebude pro vás problém daný systém prosadit. To jsme však řešili pouze otázku autorizace serveru a ne samotného obsahu elektronické pošty. Pokud si chcete být jisti, že vaši poštu nikdo číst nebude, sáhněte po kryptografickém nástroji GPG nebo PGP.

FTP

V dnešní době existuje bezpečná náhrada k FTP, a tou je například SCP. SCP využívá šifrovaného protokolu SSH, a je tudíž daleko bezpečnější než klasické FTP. Implementace SCP jako náhrada za FTP nese svá rizika. Pokud o tom uvažujete v prostředí lokální sítě, kterou znáte a ovládáte, není to takový problém. Pokud uvažujete o náhradě FTP na serveru, jenž je veřejně přístupný, nese to určitá rizika, zejména v oblasti klientů, které neznáte a neznáte ani jejich požadavky. Ještě lze říci, že náhrada FTP serveru je vhodnější tam, kde jsou k dispozici citlivá data, než na serverech, které nenabízejí nic tajného a mají vysoký bandwidth.

HTTP

Protokol HTTP má svou bezpečnou náhradu v protokolu HTTPS. Ten využívá pro šifrování dat SSL a celou komunikaci tak činí mnohem bezpečnější. Implementace SSL je celkem snadná a výrazně zlepší, zkvalitní a učiní důvěryhodnějším vaše webové sídlo.

„Bezpečné protokoly“

Nyní se podíváme na protokoly, které využívají při přenosu dat šifrování a další bezpečné metody, a můžeme je tudíž označit jako bezpečné. Půjde zejména o SSH, SSL a Kerberos. Popíšeme si, na jakém principu fungují, kde se s nimi můžeme setkat, na co si je třeba při jejich používání dávat pozor. Začneme protokolem SSH.

SSH

Secure Shell alias SSH je založen na principu šifrování veřejným klíčem. To v praxi znamená, že pracuje s dvojicí klíčů – soukromým a veřejným. V praxi se nejčastěji setkáme s volnou implementací SSH protokolu OpenSSH, vyvíjenou v rámci projektu OpenBSD. OpenSSH je dnes standardem ke komunikaci mezi unixovými systémy. Najdete jej v instalaci všech(?) linuxových distribucí, v xBSD implementacích a dalších mnoha UNIX-like systémech. Ve Windows, a to ani ve verzi 2000 a XP, není žádný šifrovaný protokol standardně implementován a implicitním nástrojem pro vzdálenou správu (hned po vašem autě) je klasický telnet, se všemi jeho nevýhodami.

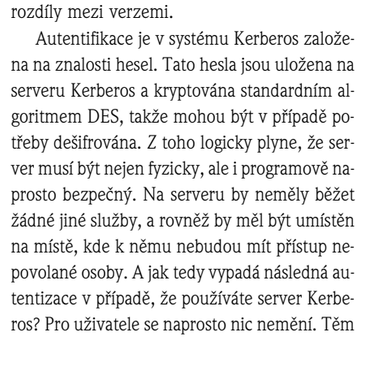
Instalace a práce s SSH je velmi snadná. Nejprve musíme nainstalovat serverovou část. Záleží na tom, jak program instalujete. Pokud například používáte Linux RedHat, nabídnou se vám ze standardní instalace k nainstalovaní soubory openssh-client, openssh-server a openssh. Pokud používáte jinou distribuci nebo instalujete ze zdrojového kódu, pravděpodobně si stáhnete jen jeden balíček a ten vám nainstaluje všechny potřebné součásti. Po nainstalování máte jako klient k dispozici tyto tři nástroje: slogin, scp a ssh. Více o jednotlivých nástrojích se dozvíte z jejich manuálových stránek.

Další věcí, kterou by měl každý uživatel počítače udělat, je vygenerování svého privátního a veřejného klíče. To lze učinit pomocí příkazu ssh-keygen. Globální konfigurační soubor pro klienty se nachází typicky v adresáři /etc/ssh/ssh_config a konfigurační soubor pro server v souboru /etc/ssh/sshd_config. Doporučuji řádně přečíst dokumentaci předtím, než začnete ssh používat, a dát si pozor na změny klíčů, které často svědčí o něčem nestandardním. Pokud se vám zobrazí hláška o změně klíče, nepište slepě, že si přejete klíč přijmout, ale raději si zkontrolujte systém fyzicky, pokud k němu máte přístup, případně kontaktujte svého systémového administrátora a na případnou změnu ho upozorněte.

SSL

SSL (*Secure Socket Layer)* je podobně jako SSH založeno na kryptografii s použitím veřejných klíčů. SSL můžete stejně dobře (stejně nejspíš ne) implementovat do webového serveru Apache, stejně jako do serveru IIS. Pokud používáte webový server Apache z nějaké standardní distribuce, budete mít pravděpodobně zabudovánu podporu SSL přímo do kódu programu. Stačí ji pak pouze aktivovat v konfiguračním souboru httpd.conf a nechat takto změněný konfigurační soubor serverem načíst. Většina direktiv je velmi dobře okomentována a vysvětlena. Pokud instalujete Apache ze zdrojového kódu, budete muset pomocí direktiv přidat podporu SSL. Více vám napoví příkaz *./configure -help*. Po nainstalování budete muset ještě vygenerovat certifikáty pro váš server. Tyto certifikáty musí být podepsány certifikační autoritou. Pokud nechcete platit za podpis certifikátu, který jste si vytvořili jen pro vlastní testovací účely, můžete si jej sami podepsat. Počítejte ale s tím, že prohlížeč vás bude při každém otevření certifikátu upozorňovat na to, že jde o samopodepsaný certifikát.

Stejně jako u SSH, se i zde můžete stát obětí útoku. Aby byl útočník v napadení systému úspěšný, bude potřebovat vaši spolupráci. Pokud se vám tedy někdy zobrazí varovná hláška o tom, že byl certifikát změněn, okamžitě věc prošetřete a neklikejte bezhlavě na OK.



KERBEROS

Kerberos je velmi zajímavý projekt, který je určen ke zvýšení bezpečnosti, a proto se na něj podíváme podrobněji a vysvětlíme si jeho princip a funkce.

Počátky systému Kerberos sahají až do roku 1983, kdy se na *MIT (Massachussets Institute of Technolgy)* ve spolupráci s firmami IBM a Digital začalo pracovat na projektu Athena, jehož cílem bylo zapojení počítačů do sítě a jejich následné využití k výuce. Vývojáři systému narazili během vývoje na mnoho problémů, zejména s bezpečností dat a autentizací. Výsledek byl systém Kerberos, který pracuje jako síťový autentifikační systém. V současné době je Kerberos k dispozici ve dvou hlavních verzích, a to ve verzi 4 a 5 (přesněji IV a V). My si nejdříve popíšeme, na jakém principu Kerberos pracuje, a pak se podíváme na rozdíly mezi verzemi.

Autentifikace je v systému Kerberos založena na znalosti hesel. Tato hesla jsou uložena na serveru Kerberos a kryptována standardním algoritmem DES, takže mohou být v případě potřeby dešifrována. Z toho logicky plyne, že server musí být nejen fyzicky, ale i programově naprosto bezpečný. Na serveru by neměly běžet žádné jiné služby, a rovněž by měl být umístěn na místě, kde k němu nebudou mít přístup nepovolané osoby. A jak tedy vypadá následná autentizace v případě, že používáte server Kerberos? Pro uživatele se naprosto nic nemění. Těm se zobrazí standardní přihlašovací dialog typu login: a password:. Důležité je, jak vnitřně Kerberos pracuje, a co z toho pro uživatele plyne. Zde rovněž vyvstávají rozdíly mezi verzemi IV a V. Podíváme se na obě varianty.

KERBEROS IV.

Po zadání jména (login:) odešle stanice serveru Kerberos zprávu s vaším uživatelským jménem a informací o tom, že se chcete přihlásit do systému. Server, jelikož zná vaše heslo a jméno, ověří vaše uživatelské jméno ve své databázi, a pokud je v pořádku, odešle vám tiket, jejž zašifruje vaším heslem, které také zná. Když tento tiket dorazí do vašeho systému, požádá vás systém o zadání hesla. Pokud zadáte správné heslo, dešifruje jím systém tiket a přihlásí vás do systému. V momentě, kdy vám systém povolí přihlášení, zlikviduje informaci o vašem heslu a dále výhradně pracuje s tiketem.

KERBEROS V.

V této verzi serveru Kerberos čeká stanice až do doby, kdy zadáte své heslo. Až poté kontaktuje server a odešle mu zprávu s vaším uživatelským jménem a aktuálním časem, který je šifrován vaším heslem. Server po obdržení této zprávy zjistí ve své databázi vaše jméno a heslo, a pokusí se dešifrovat časový údaj. Pokud vše proběhne v pořádku, pošle vám server tiket zašifrovaný vaším heslem.

Z informací tedy v praxi plyne následující:

- Hesla nejsou uložena na žádné stanici v síti kromě serveru Kerberos
- Hesla se při procesu autentizace nepřenáší po síti, pracuje se výhradně s tikety.

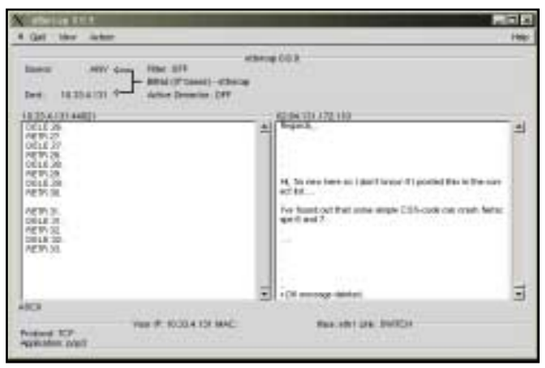
IMPLEMENTACE KERBERA DO STÁVAJÍCÍ SÍTĚ

Implementace systému Kerberos není jednoduchou záležitostí. Je třeba si rozmyslet všechny potřeby sítě, vzít v potaz každý detail. Ne pro každého je použití systému Kerberos vhodným řešením. Samotná instalace již tak složitá není.

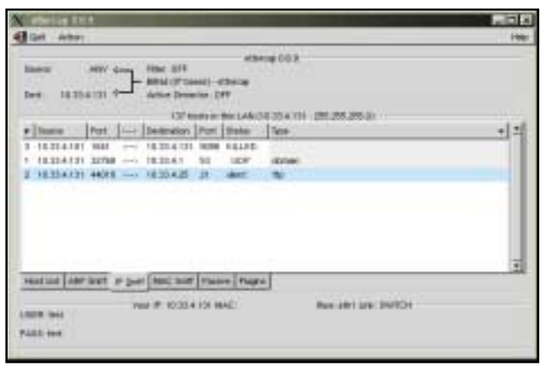
ZÁVĚREM

Na závěr lze říci, že používání autentizačních mechanismů není snadná věc. Pokud to s bezpečností myslíte vážně, musíte začít uvažovat o nahrazení standardních mechanismů jejich bezpečnějšími alternativami. Proces přechodu na bezpečnější alternativy není jednoduchou záležitostí a vyžaduje mnoho změn současného systému. Tento přechod se vám však bohatě vyplatí a nakonec bude sloužit ke spokojenosti správců sítě a uživatelů.

Nápady, dotazy, návrhy témat, jež vás zajímají a připomínky zasílejte na adresu igm@centrum.cz. Na vaše dotazy k této problematice se pokusíme najít odpovědi. 3 0163/FEL ❑



▲ Citlivé informace zasílané e-mailem by se neměly dostat do nepovolaných rukou



▲ Vidíte-li při přenosu z FTP serveru své jméno a heslo, nejde o bezpečný přenos