

ALWIL Software

iAus

dokumentace pro administrátory

Obsah

1	Co je iAVS?	7
2	Jak to funguje?	9
3	Bezpečnost?	11
4	Zrcadlení	13
	4.1 Jak zrcadlit - obecně	13
	4.2 Pravidelná údržba - obecně	13
	4.3 Jak zrcadlit - hotové řešení	14
	4.4 Pravidelná údržba - hotové řešení	14
	4.5 Nastavení klientů iAVS	15
5	Jak se vybírá server?	17
6	Soubor servers.def	19
7	Soubor groups.def	21
8	Soubory index.*.def	23
9	Soubor packages.def	25
10	Soubory *.stamp	27
11	Soubor mirror.def	29
12	Soubor lock.def.stamp	31
13	Soubor syncer.ini	33
14	Soubory s popisem distribuce	35
15	Příkaz at	37
16	Program tmcr.exe	39
17	Program mirror.pl	41
18	Program cldistro.pl	43
19	Program mkdistro.pl	45
20	Program chdistro.pl	47
21	Program uploader.pl	49
22	Program doall.pl	51
23	Program md5.exe	53

24	Změny	55
25	Často kladené dotazy	57
26	Často kladené dotazy - Perl	59
27	Často kladené dotazy - Unix	61
28	Poděkování	63

Seznam obrázků

1 Co je iAVS?

iAVS je systém zabezpečující diferenční aktualizace software. V současné době je použit pro diferenční aktualizace souboru VPS.

Ke změně aktualizacího procesu došlo především kvůli tomu, že měsíční aktualizace antivirové databáze byly příliš 'pomalé' vzhledem k rychlosti, jakou jsou schopny se šířit nové 'internetové' viry. Stávající systém aktualizací se stal pro svoji nepružnost nevyhovujícím.

Původní mechanismy aktualizace prozatím zůstávají zachovány, viz dokumentaci Avast32 a změnový dokument AUPD.TXT.

Předem upozorňujeme, že tento manuál je určen k rychlé orientaci a proto v hojné míře využívá odkazů. Není proto příliš vhodný pro lineární čtení. Je určen především pro správce velkých sítí.

2 Jak to funguje?

Po spuštění programu syncer.exe, ať již přímo či z klienta Avast32, se stanou následující věci:

Z lokálního souboru servers.def, který obsahuje seznam serverů, se vybere server. (Jak se to děje, se dá dozvědět v této kapitole).

Z tohoto serveru se stáhne případný novější soubor servers.def a znovu se z něj vybere server.

Pokud je na serveru soubor groups.def s definicemi skupin počítačů, stáhne se jeho obsah a podle daných pravidel se vybere skupina, do které patří tento počítač.

Stáhne se soubor index.*.def podle jména skupiny. Tento soubor určuje, které balíky se budou instalovat.

Stáhne se soubor packages.def s definicemi balíků a jejich závislostí.

Začnou se interpretovat soubory index.*.def a packages.def a začne se provádět aktualizace balíků. Pokud aktualizuje program sám sebe, může v tomto okamžiku dojít k restartu celého procesu.

Proces se ukončí. Program si vyjíměčně požádá o restart systému. (Nikdy si o něj nepožádá kvůli aktualizaci databáze virů.)

Poznámka: všechny soubory *.def se stahují jen v případě, že na lokálním disku neexistují, popř. mají jiné datum modifikace než soubory na serveru.

3 Bezpečnost?

Ano. :-)

Každý binární soubor, který je stažen na lokální počítač, je zkontrolován, zda má:

- stejnou délku
- stejný MD5 checksum - 128 bitů
- platný podpis ALWIL Software - 320 bitů

Použitý algoritmus pro vytváření digitálního podpisu má privátní klíč o délce 1024 bitů. Daná šifra je asymetrická a tudíž není prakticky možné bez znalosti privátního klíče zfalšovat podpis a podstrčit tak na klientský stroj jiný soubor, než jsme zamýšleli. Formát binárních souborů je ještě vnitřně chráněn a není dokumentován. K tomu se ještě přidává to, že i naše webové servery jsou chráněny.

Ještě jednou, pro shrnutí, kroky, které by musel provést potenciální zlosyn:

- zjistit formát našich binárních souborů a vyrobit takový platný soubor - nesnadné
- vyrobit pro něj MD5 checksum - velmi snadné
- vyrobit pro něj podpis ALWIL Software - téměř nemožné
- vyrobit soubor package.def - snadné
- umístit na nějaký server - záleží na ochraně serveru

4 Zrcadlení

Zrcadlení znamená, že si na své lokální servery umístíte kopie (zrcadlo, mirror) našich souborů.

Pro vás, jako správce velké sítě je rozhodně výhodnější si takové zrcadlo udržovat. Aktualizace z vašich serverů bude rychlejší, spolehlivější a dá vám možnost kontrolovat verze balíků, které se instalují na cílové stanice.

V dalších podkapitolách je popsán jak obecný postup, tak i automatizované a pohodlné řešení pomocí sady programů napsaných v Perlu.

4.1 Jak zrcadlit - obecně

Nejprve je třeba si vytvořit adresář, kam se budou pravidelně kopírovat novější soubory z našeho distribučního adresáře. Pak do něj stáhněte všechny soubory. Je vhodné mít nějakou oblíbenou utilitu, která to zvládne; ideální by bylo, kdyby stahovala jen novější soubory. URL na naše servery jsou k nalezení v souborech servers.def.

Pak si vytvořte soubory:

- servers.def s nastavením vašich serverů.
- groups.def s rozdělením počítačů do skupin.
- všechny soubory index.*.def s výběrem balíků podle vašich skupin.

Soubory groups.def a index.*.def vytvářejte pouze v tom případě, že chcete omezovat instalaci některých balíků. Pokud chcete mít na všech počítačích instalaci shodnou s tou, která je na našich serverech, můžete dané soubory vynechat - budou se používat námi definované.

Pokud byste čistou náhodou umísťovali soubory na HTTP server, je třeba pro ně vytvořit soubory *.stamp.

Vytvořte distribuční adresář, který vznikne zkopírováním souborů ze zrcadla našeho serveru a připravených *.def souborů.

Pak nahrajte všechny soubory na vaše distribuční servery.

Hotové řešení v Perlu je popsáno [zde](#).

Pravidelná údržba je popsána [zde](#).

4.2 Pravidelná údržba - obecně

V podstatě záleží na vás. Je možné, že občas asi pocítíte potřebu změnit soubory servers.def nebo groups.def.

Určitou výjimku tvoří soubory `index*.def`, do kterých budeme časem doplňovat další balíky. Vy byste pak měli aktualizovat váš soubor `index*.def`.

Jinak je pravidelná údržba nenáročná - stačí zrcadlit naše soubory a kopírovat výsledný distribuční adresář na vaše distribuční servery.

4.3 Jak zrcadlit - hotové řešení

Toto řešení je napsáno v Perlu. Nahlédněte do jeho často kladených otázek.

Rozbalte distribuci utilit i s podadresáři.

Do podadresáře `mirror` umístěte soubor `servers.def` z našeho serveru. (Je například součástí instalace.)

Nastavte si konfigurační soubor `mirror.cfg` v adresáři `cfgs`. Zkuste si spustit program `mirror.pl`. Ten by měl do adresáře `mirror` stáhnout poslední možný stav souborů. V adresáři `logs` si můžete zkontrolovat případná chybová hlášení (soubory `mirrorDATUM.log`).

Pak si vytvořte soubory:

- `servers.def` s nastavením vašich serverů.
- `groups.def` s rozdělením počítačů do skupin.
- všechny soubory `index*.def` s výběrem balíků podle vašich skupin.

a umístěte je do adresáře `deffiles`.

Soubory `groups.def` a `index*.def` vytvářejte pouze v tom případě, že chcete omezovat instalaci některých balíků. Pokud chcete mít na všech počítačích instalaci shodnou s tou, která je na našich serverech, můžete dané soubory vynechat - budou se používat námi definované.

Před vytvořením nové distribuce je dobré starou distribuci smazat; k tomu je určen program `cldistro.pl`.

Spusťte program `mkdistro.pl`. V adresáři `distrib` by tím pádem měla být vytvořena vaše distribuce.

Program, který dokáže velmi povrchně odhalit některé chyby v distribuci se jmenuje `chdistro.pl`.

Nastavte si správně všechny konfigurační soubory v adresáři `cfgs` pro program `uploader.pl` a spusťte jej. V adresáři `logs` si můžete zkontrolovat případná chybová hlášení (soubory `uploaderDATUM.log`).

V tomto okamžiku byste na všech serverech měli mít stejnou distribuci.

Pak už jen nastavit klienty a je to.

4.4 Pravidelná údržba - hotové řešení

Toto řešení je napsáno v Perlu. Nahlédněte do jeho často kladených otázek.

V podstatě záleží na vás. Je možné, že občas asi pocítíte potřebu změnit soubory `servers.def` nebo `groups.def`.

Určitou výjimku tvoří soubory `index*.def`, do kterých budeme časem doplňovat další balíky. Vy byste pak měli aktualizovat váš soubor `index*.def` (pokud jste se rozhodli omezovat maximální verze balíků). Program `mkdistro.pl` vám prozradí, jak se liší naše a vaše soubory `index*.def`.

Jinak je pravidelná údržba nenáročná - stačí zrcadlit naše soubory a kopírovat výsledný distribuční adresář na vaše distribuční servery.

Automaticky, například pomocí programu `at`, byste tedy měli pravidelně spouštět tyto programy v tomto pořadí:

```
mirror.pl  
cldistro.pl  
mkdistro.pl  
chdistro.pl  
uploader.pl
```

Všechny tyto úkoly dohromady zvládá program `doall.pl`, který provede všechny výše uvedené činnosti a zašle mail správci.

Není od věci upozornit, že program `at` neumožňuje zápis na sdílené disky, proto by se daná dávka měla autorizovat: `net use x:\\server\path /USER:DOMAIN\user password`

4.5 Nastavení klientů iAVS

Pokud se nově instaluje Avast32, stačí k instalaci přidat soubory `syncer.ini` a `servers.def`. Blíže viz dokumentace o nastavení administrátorské instalace.

Pokud je třeba změnit nastavení stávajících stanic (serverů) s Avast32, provedete běžnou aktualizaci programu pomocí vám jistě dobře známých souborů `*.tm` a `*.tmd`. Nový program `avupdate`, který je v posledních instalacích, umožňuje aktualizovat i soubory `syncer.ini` a `servers.def`. Jediné, co potřebujete, je nahrát tyto soubory do adresáře, kam běžně dáváte soubory `*.tm` a vytvořit pro ně soubory `*.tm` pomocí programu `tmcr.exe`. Blíže k tomuto viz soubor `AUPD.TXT`, přiloženém k instalaci Avast32.

5 Jak se vybírá server?

Prozatím se jedná pouze o náhodný výběr. To se v blízké budoucnosti změní (budou se dát nastavovat priority apod.)

6 Soubor servers.def

Soubor servers.def obsahuje informace o všech serverech, které obsahují adresář se všemi aktualizacími soubory. Jedná se o běžný windowsovský ini soubor.

Poznámka: Zde je dokonalá metoda, jak věci pokazit: Pokud do souboru server.def napíšete nesprávná/vadná url, dojde při aktualizaci k přepsání lokální kopie souboru servers.def a tím dojde k tomu, že všechny stroje ztratí možnost aktualizace, protože nebudou mít odkud aktualizovat.

Skupina [Servers]

count = číslo - určuje počet definic serverů v souboru.

Skupina [ServerN]

N v názvu této skupiny je číslo serveru. První server má číslo 0.

name = řetězec - jméno serveru, které se zobrazuje během aktualizace

url = řetězec - url, kde jsou všechny soubory. Musí začínat buď http:// nebo file:// a dodržovat všechny konvence pro zadávání url.

StatsCollectType = řetězec - určuje způsob, jakým se ukládají výsledky statistiky

- cgi - data se posílají pomocí HTTP requestu POST cgi skriptu

StatsCollectUrl = řetězec - pro cgi: url ke skriptu pro zpracování statistiky. Toto url by mělo (pokud ho vůbec potřebujete) vést na váš server, aby se vaše interní statistiky nemíchaly s globálními statistikami našich serverů.

Příklad:

```
[servers]
count=2
```

```
[Server0]
name=Hlavni server
url=http://www.organizace.cz/iavs
```

```
[Server1]
name=Druhy server
url=file://\server\share\iavs
```


7 Soubor groups.def

Soubor groups.def popisuje rozdělení počítačů do skupin. Formát jednoho záznamu je:

skupina=pravidla;

Skupina je buď jméno skupiny (později použito pro rozlišení souborů index.def) nebo je prázdná a v tom případě má daný počítač zakázanou aktualizaci.

Pravidla jsou specifikována takto:

proměnná:hodnota

Možné proměnné k testování:

- winname - Windows jméno počítače
- ip - IP adresa počítače
- ipname - DNS/WINS jméno počítače
- sys - operační systém počítače

winname:

ipname:

Provádí se porovnávání řetězců. Nezáleží na velikosti písmen, je možno použít wildchary ? a *.

sys:

Porovnání s následujícími hodnotami:

- 9x - Windows 9x
- nt - Windows NT
- 95 - Windows 95
- 98 - Windows 98
- nt4 - Windows NT 4
- 2000 - Windows 2000

ip:

Buď přesné číselné porovnání (192.168.1.243) nebo 'maskové' porovnání (adresa/bitmask např. 192.168.1.0/24 pro 'céčkový' subnet).

Je možno použít tyto operátory (asociují se zleva, logické and má vyšší prioritu než logické or):

- () - určení pořadí provádění
- & - logické and
- and
- | - logické or
- or
- ! - logické not
- not

Vyhodnocuje se shora dolů. Porovnávání se ukončí již při první shodě. Pokud nedojde k žádné shodě, je takový počítač automaticky zařazen do skupiny all.

Příklad 1:

```
bleedingedge=( ip:192.168.11.0/24 | ip:192.168.12.0/24 ) & sys:nt;  
nt=sys:nt;  
others=winname:*;
```

Z tohoto plyne, že se vyskytují pouze tři skupiny (all nebude, protože poslední pravidlo 'sežere' všechny nezařazené). První skupinu tvoří pouze ntčkáři ze subnetů 192.168.11 a 192.168.12. Druhou skupinu pouze ntčkáři, třetí skupinu všichni ostatní.

Příklad 2:

```
vip=winname:boss;  
vip=winname:boss_secretary;  
vip=winname:manager1;
```

Zde jsou dvě skupiny. V první skupině jsou zařazeni pouze 'speciální' uživatelé, ostatní propadnou do skupiny all.

Příklad 3:

```
obchodni=ip:192.168.11.0/24 | ip:192.168.10.234 | ip:192.168.10.235 | ip:192.168.10.236
```

```
obchodni=ip:192.168.11.0/24  
obchodni=ip:192.168.10.234  
obchodni=ip:192.168.10.235  
obchodni=ip:192.168.10.236
```

Obě možnosti uvedené výše jsou ekvivalentní. Spodní výraz je daleko přehlednější.

Příklad 4:

```
common=ipname:*.obchodni.domena.cz;  
vip=ipname:reditel.obchodni.domena.cz;
```

Možný problém může nastat, když se žravější pravidlo použije příliš brzy. Jak je vidět výše, ke skupině vip nikdy nemůžeme dojít, protože dojde ke shodě již na první podmínce.

Pokud soubor groups.def neexistuje, automaticky se pro všechny používá skupina all.

8 Soubory index.*.def

Soubory index.*.def popisují balíky a způsob, jak se instalují. Místo hvězdičky se dosazuje jméno skupiny, tak, jak je specifikováno v souboru groups.def.

Každý řádek má formát:
příkaz:balík;

nebo:
příkaz:balík verze;

Příkaz:

- install - provede instalaci balíku nezávisle na tom, zda již na stanici existuje předchozí verze
- update - provede aktualizaci balíku
- ignore - pro balík neprovádí žádné akce

Balík: jméno balíku, tak, jak je specifikováno v souboru packages.def. V současné době pouze balík vps32.

Verze: nepovinný parametr určující, na jakou maximální verzi se má aktualizovat. To může být vhodné, pokud jsou například počítače rozděleny do dvou skupin: jedna pro běžné uživatele, kde běží stabilní a odzkoušená konfigurace, a druhá pro pokročilé (otrlé) uživatele, kteří používají nejnovější verze.

Každý soubor index.def má implicitní první řádek, který nelze ovlivnit. Je to řádek install syncer;, který zabezpečuje, že aktualizací program aktualizuje jako první sám sebe.

9 Soubor packages.def

Soubor packages.def obsahuje pravidla pro provádění aktualizací. Do tohoto souboru byste neměli zasahovat. Jeho formát se bude v budoucnosti často měnit. Pro výběr instalovaných balíků existují soubory index.*.def.

Formát je následující:

```
packagedef( jméno_balíku pravidla_pro_rozpoznání_balíku )  
  
nová_verze: stará_verze  
  
příkaz1;  
příkaz2;  
...
```

Ještě jednou: do tohoto souboru byste neměli zasahovat. Protože příkazy nejsou specifikovány v žádné veřejné dokumentaci, každý zásah může vést k nefunkčnosti systému.

10 Soubory *.stamp

Soubory *.stamp existují pro každý soubor *.def (celé jméno je tedy *.def.stamp). Obsahují pouze časový údaj vzniku souboru *.def. Formát je dekadické číslo vyjadřující počet sekund od 1.1.1970 (epoch). (Pro C programátory: time_t);

Před každým stažením souboru *.def se nejdříve stáhne odpovídající soubor stamp a ověří se, zda je jeho časový údaj shodný se souborem již existujícím na lokálním disku. Pokud jsou shodné, pracuje se dále s dříve staženým souborem def. Šetří to čas, pokud se soubory na serveru nezměnily.

11 Soubor mirror.def

Soubor mirror.def obsahuje data pro zrcadlicí (mirrorovací) programy. Struktura je jednoduchá, jedná se o tři sloupce oddělené čárkami na řádku. Jsou to: jméno souboru, délka souboru a MD5 hash.

Tento soubor používá utilita mirror.pl. Pro běh vlastního iAVS není přímo potřebný.

12 Soubor lock.def.stamp

Soubor lock.def.stamp se může na zdrojovém serveru vyskytnout pouze v případě, že zrovna probíhá aktualizace souborů. Jeho formát je stejný, jako u ostatních razítkových souborů. Jeho časový údaj označuje počátek aktualizace serveru. Pokud je stáří tohoto souboru větší než je únosná míra (toto je velmi relativní), je možné, že při aktualizaci serveru došlo k havárii a proto zůstal 'zamčený'. Doporučuje se v tomto případě informovat správce serveru.

13 Soubor syncer.ini

V souboru syncer.ini je nastavení klienta. Toto nastavení je uloženo na stanici, ve stejném adresáři, kde se nachází syncer.exe. Měl by být nastaven správcem sítě a distribuován pomocí stávající technologie aktualizací.

Skupina [Syncer]

Mode = číslo - určuje typ zobrazení okna.

- 0 - velké okno s tlačítkem Zrušit
- 1 - velké okno s tlačítkem Zrušit, na konci se samo uzavře
- 2 - velké okno bez tlačítka Zrušit
- 3 - okno má jen hlavní teploměr (progress bar)
- 4 - pouze ikona na liště (v trayi) bez možnosti zvětšení
- 1000 - žádné okno

Lang = číslo - určuje jazyk.

- %Avast32% - používá stejné nastavení jako Avast32
- 0409 - anglicky
- 0405 - česky

HttpProxy = řetězec - určuje adresu a port HTTP proxy serveru. Bude ignorován, pokud je zadáno SocksProxy

- proxy:port - plné určení proxy
- %Avast32% - používá stejné nastavení jako Avast32

SocksProxy = řetězec - určuje adresu a port SOCKS proxy serveru.

- proxy:port - plné určení proxy
- %Avast32% - používá stejné nastavení jako Avast32

SocksUserId = autentifikační řetězec pro SOCKS proxy. Bude ignorován, pokud není zadáno SocksProxy

HideInAvast = číslo - určuje, zda se v Avast32 nezobrazuje tlačítko pro aktualizaci.

- 0 - zobrazuje
- 1 - nezobrazuje

DoNotAskForReboot = číslo - určuje, zda se klient nesmí zeptat na přestartování systému.

- 0 - smí
- 1 - nesmí

Příklad:

```
[Syncer]
mode=1
lang=0405
```

Avast32 Antivirus

httpproxy=proxy.organizace.cz:8080

14 Soubory s popisem distribuce

Ve verzi 1.10 byla přidána možnost pracovat s více různými distribucemi. Je samozřejmě možné pracovat dále s jednou distribucí. Není potřeba zadávat její název. Její jméno je 'default', def adresář je 'deffiles' a dir adresář je 'distrib', takže je zachována kompatibilita s verzí 1.00.

Soubory mají jméno nazev.dst a nahrávají se do adresáře cfgs.

name = řetězec - jméno dané distribuce, pouze pro zobrazení.
def = řetězec - ze kterého adresáře se berou soubory *.def. Relativní cesta k adresáři se skripty.
dir = řetězec - kam se nahrává výsledek (programem mkdistrib.pl). Relativní cesta k adresáři se skripty.

Příklad:

```
name=první distribuce  
def=defsr  
dir=distr
```

Tento soubor se jmenuje první.dst, je v adresáři cfgs a z utilit se na něj odkazuje jako první.

15 Příkaz at

Příkaz at je jednoduchý plánovač úloh běžící v prostředí WindowsNT. Je standardní součástí WindowsNT. Aby se dal použít, musí běžet služba Schedule. Spusťte Ovládací panely a z nich applet Služby. Najděte v seznamu službu Schedule. Pokud není spuštěna, změňte režim spouštění na Automatický, a službu spusťte. Pokud jste službu Schedule vůbec nenalezli, máte pravděpodobně nainstalován MS Task Scheduler, který službu Schedule nahrazuje. Hledejte službu Task Scheduler a zajistěte její spouštění.

Jestliže běží příslušná služba, můžete z příkazové řádky spustit program at. Vypíše všechny naplánované úlohy. Pokud Vás zajímá nápověda, zkuste:
at /?

První spuštěná úloha

Podívejme se nyní na to, jak se spouští úlohy. Pro začátek vytvoříme jednoduchý dávkový soubor run.bat v adresáři C:\.

```
run.bat:  
dir > C:\run.out
```

Spuštění naplánujeme takto:

```
at 16:30 c:\run.bat
```

V 16:30 vznikne soubor c:\run.out. Jeho obsahem je obsah adresáře ve kterém byl program run.bat spuštěn. Jednoduchou analýzou zjistíme, že to byl adresář C:\winnt\system32.

Máme rádi okénka

Systém WindowsNT je pokročilý operační systém, který umožňuje i tvorbu aplikací, které s uživatelem komunikují jinak než prostřednictvím příkazové řádky. Vyzkoušíme si naplánování spuštění takového programu.

```
at 10:00 notepad.exe
```

A v deset hodin nic. Prozkoumáním seznamu úloh v Task Manageru zjistíme, že notepad se spustil, ale vidět není. Je to tím, že se spustil pod systémovým účtem a přihlášený uživatel se k němu nedostane. Pro plánování tedy zapomeňme na jakékoliv interaktivní úlohy. Programy spuštěné příkazem at se nesmí na nic ptát a musí samy skončit.

Máme rádi síť

Úlohy naplánované pomocí programu at nemohou přistupovat na síťové disky, ať už namapované nebo pomocí UNC. Naplánované úlohy jsou totiž spuštěny pod lokálním systémovým účtem, který nemá právo přistupovat na síť. S tím je třeba při návrhu plánovaných aplikací počítat.

Periodické spouštění

Tak tohle pro změnu jde. Pomocí přepínače /every: lze specifikovat dny v týdnu nebo měsíci, ve kterých se má úloha spustit. Příklady:

```
at 14:00 /EVERY:m,t,w,th,f,s,su c:\run.bat
```

spustí úlohu každý den ve 14:00.

```
at 14:00 /EVERY:15 c:\run.bat
```

spustí úlohu vždy 15. v měsíci ve 14:00.

Pro označení dnů v týdnu se používá první písmeno z anglického názvu dne, pro některé z nich pak první dvě písmena. Pondělí: m, úterý: t, středa: w, čtvrtek: th, pátek: f, sobota: s, neděle: su. Pro označení dne v měsíci se používá jeho pořadové číslo od začátku měsíce.

16 Program tmcr.exe

Program tmcr.exe slouží k vytvoření souboru s časovým razítkem. Tento soubor se potom použije pro distribuci souborů pomocí mechanismu automatických aktualizací pomocí tmd. Příklady:

tmcr.exe syncer.ini

Po spuštění tohoto programu vznikne v aktuálním adresáři soubor syncer.tm.

Chcete-li distribuovat vlastní soubory syncer.ini a servers.def na stanice, vytvořte k nim soubory syncer.tm a servers.tm a všechny je nahrajte do adresáře, do kterého ukládáte soubory pro aktualizaci programu Avast32 (soubory sp32.tm a sp32.tmd).

17 Program mirror.pl

Popis:

Program mirror.pl je určen k provádění automatických aktualizací do adresáře mirror. Podporované protokoly jsou http:// a file://.

Program vytváří logy do adresáře logs. Jméno každého souboru je mirrorYYYY-MMDD_HHMMSS.log.

Závislosti:

program md5.exe - je součástí dodávky

modul LWP - v ActiveState distribuci Perlu se tento modul instaluje automaticky

Parametry příkazové řádky:

-singlelog - výsledný log má jméno mirror.log

Návratové hodnoty:

-2 - některé soubory se nezkopírovaly

-1 - fatální chyba

0 - nepřeneseny žádné soubory, zdrojový server se nezměnil

1 - přeneseny aspoň nějaké nové soubory

Konfigurační soubory:

V adresáři cfgs je soubor mirror.cfg.

retry_count = číslo - určuje počet pokusů v případě, že se nepodařilo stáhnout všechny soubory. (Default: 3)

retry_interval = číslo - čas v minutách, po který program čeká mezi jednotlivými pokusy. (Default: 5)

http_proxy = řetězec - adresa http proxy ve tvaru http://server:port.

company = řetězec - jméno společnosti. Používá se kvůli identifikaci problémů s mirroringem.

company_admin = řetězec - email na administrátora systému iAVS.

Změna proti verzi 1.00: nyní je tato identifikace společnosti/administrátora povinná, bez ní program nepoběží.

18 Program cldistro.pl

Popis:

Tento program pouze vymazává obsah adresáře distrib.

Závislosti:

Žádné.

Parametry příkazové řádky:

Žádné nebo název distribuce.

Návratové hodnoty:

Nedefinované.

19 Program mkdistro.pl

Popis:

Tento program vytvoří pro všechny soubory *.def v adresáři deffiles soubory *.stamp a všechny tyto soubory zkopíruje do adresáře distrib. Do tohoto adresáře rovněž zkopíruje všechny zbývající soubory z adresáře mirror.

Navíc je schopen zjistit rozdíl mezi našimi a vašimi soubory index.*.def.

Program vytváří log soubor mkdistro.log do adresáře logs.

Závislosti:

program md5.exe - je součástí dodávky

modul File::Copy - v ActiveState distribuci Perlu se tento modul instaluje automaticky

Parametry příkazové řádky:

Žádné nebo název distribuce.

Návratové hodnoty:

-1 - fatální chyba

0 - vše je v pořádku

1 - alespoň jeden možný problém se soubory index.*.def

20 Program chdistro.pl

Popis:

Tento program testuje adresář distrib na přítomnost některých možných chyb.

Pořadí testů:

- pro všechny skupiny ze souboru groups.def musí existovat soubory index.*.def.
- všechny balíky (packages), které jsou referencovány ze všech souborů index.*.def, musí být definovány v souboru packages.def.
- pro všechny soubory, které jsou uvedeny v souboru mirror.def, musí platit, že mají odpovídající délky a kontrolní součty.

Program vytváří logy do adresáře logs. Jméno každého souboru je chdistroYYYY-MMDD_HHMMSS.log.

Závislosti:

program md5.exe - je součástí dodávky

Parametry příkazové řádky:

-singlelog - výsledný log má jméno chdistro.log
je možné zadat název distribuce.

Návratové hodnoty:

-1 - fatální chyba
0 - vše v pořádku

21 Program uploader.pl

Popis:

Tento program nahrává obsah adresáře distrib na distribuční servery. Používá buď FTP přenos nebo kopírování souborů.

Program vytváří logy do adresáře logs. Jméno každého souboru je uploaderYYYY-MMDD_HHMMSS.log.

Program vytváří soubory *.upl do adresáře cfgs. Tyto soubory obsahují poslední přenesený stav souborů.

Závislosti:

program md5.exe - je součástí dodávky

modul Net::FTP - není součástí standardní instalace ActiveState. Tento modul je potřeba pouze pro aktualizaci pomocí FTP. Instaluje se tak, že se z adresáře, kde je perl.exe, spustí program ppm.bat, napíše se install libnet a pak se jen odpovídá na dotazy. Podmínkou je připojení k internetu.

modul File::Copy - je součástí standardní dodávky Perlu od ActiveState.

Parametry příkazové řádky:

-singlelog - výsledný log má jméno uploader.log

-all - aktualizuje všechny soubory, ne jen změny.
je možné zadat název distribuce.

Návratové hodnoty:

-1 - fatální chyba

0 - nepřeneseny žádné soubory, cílový server je synchronizován

1 - přeneseny alespoň nějaké soubory.

Konfigurační soubory:

Pro běh tohoto programu je třeba mít nastaveny konfigurační soubory v adresáři cfgs:

Konfigurační soubor uploader.cfg

Zde jsou pouze dva zajímavé parametry:

dry_run = číslo - zda program běží nasucho, nebo něco dělá.

- 0 - funguje normálně
- 1 - jen vypisuje, co by dělal

servers = řetězec - názvy souborů *.srv, oddělené čárkami. Například: servers=srv1,srv2 bude provádět aktualizace serverů, tak jak jsou popsány v souborech srv1.srv a srv2.srv.

Pokud se používá více distribucí, místo jednoho řádku 'servers' se použije pro každou distribuci řádek 'servers_nazevdistra'. Například pro distribuci 'prvni' se použije přiřazení do 'servers_prvni'.

Konfigurační soubory serverů *.srv

Parametry:

type = řetězec - typ uploadu

- ftp - provádí se upload pomocí ftp
- file - pouze se kopírují soubory

dest_path = řetězec - cesta, kam umístit soubory (platí pro ftp i file)

retry_count = číslo - určuje počet pokusů v případě, že se nepodařilo poslat všechny soubory. (Default: 3)

retry_interval = číslo - čas v minutách, po který program čeká mezi jednotlivými pokusy. (Default: 5)

ftp_site = řetězec - adresa ftp serveru (platí jen pro ftp)

ftp_port = číslo - port ftp serveru (platí jen pro ftp)

ftp_firewall = řetězec - adresa ftp proxy (platí jen pro ftp)

ftp_login = řetězec - přihlašovací jméno na ftp server (platí jen pro ftp)

ftp_password = řetězec - heslo pro přístup na ftp server (platí jen pro ftp)

Příklad 1:

```
type=file  
dest_path=x:\cesta\avast
```

Příklad 2:

```
type=ftp  
ftp_site=ftp.organizace.cz  
ftp_login=uploader  
ftp_password=heslo  
dest_path=/pub/avast/iavs
```

22 Program doall.pl

Popis:

Tento program pouze volá všechny ostatní. Stará se o jejich návratové hodnoty a vytváří o tom zprávu.

Log se vytváří do adresáře logs pod jménem doall.log.

Závislosti:

Závisí na všech ostatních programech.

modul Net::SMTP - není součástí standardní instalace ActiveState. Tento modul je potřeba pouze pro zasílání mailů. Instaluje se tak, že se z adresáře, kde je perl.exe, spustí program ppm.bat, napíše se install libnet a pak se jen odpovídá na dotazy. Podmínkou je připojení k internetu.

Musí mít nastaven v cestě (PATH) program perl.exe.

Parametry příkazové řádky:

Žádné.

Návratové hodnoty:

-1 - fatální chyba

0 - vše v pořádku.

Konfigurační soubory:

V adresáři cfgs je soubor doall.cfg.

distros = řetězce oddělené čárkou - obsahuje jména všech potřebných souborů s informacemi o distribuci.

smtp_server = řetězec - jméno SMTP serveru pro odchozí poštu.

smtp_port = číslo - port, na kterém běží SMTP server.

mail_to = řetězec - emailová adresa, kam mail posílat.

mail_from = řetězec - nepovinný parametr, nastavuje se, když server vyžaduje platné jméno odesílatele.

dont_send_positive_mails = číslo - pokud všechny volané programy vrátí 0 jako návratovou hodnotu, nepošle žádný mail

- 0 - posílá maily vždy
- 1 - posílá maily pouze, když se stalo něco zajímavého

23 Program md5.exe

Příkazová řádka:
md5.exe soubor

Vrátí na konzoli MD5 checksum souboru.

Program je interně používán některými našimi utilitami.

24 Změny

Tato stránka popisuje změny proti předchozí verzi tohoto dokumentu.

Verze 1.00

- První verze.

Verze 1.10

- Přidána možnost práce s více cílovými distribucemi.
- Odkoušena práce pod Unixem. FAQ.
- Povinná identifikace administrátora. Viz soubor mirror.cfg.
- Opravy různých malých chybek.

25 Často kladené dotazy

Malé FAQ týkající se Perlu je zde.

Q:Perl a příkazová řádka se nám nelíbí. Budou utility s grafickým rozhraním?

A: Asi vás musíme zklamat, ale v nejbližší budoucnosti se s tím nepočítá. Vzhledem k tomu, že formáty a algoritmy pro iAVS jsou v tomto manuálu popsány dostatečně, nic vám nebrání v sepsání vlastních programů.

26 Často kladené dotazy - Perl

Protože většina dodatečných utilit pro iAVS, které mají ulehčit správu administrátorům, je napsaná v Perlu, zde je několik otázek a odpovědí.

Malé FAQ:

Q:Co je to Perl?

A:Perl je buď to, co již dávno používáte a proto tuto kapitolu můžete přeskočit, nebo to, co si rozhodně musíte zamilovat. :-)

Jedná se o velmi rozšířený skriptový jazyk, široce používaný hlavně ve světě UNIXových systémů. Pokud znáte shell script, awk, sed a C, nebudete s Perlem mít téměř žádné problémy. Ani v opačném případě byste je mít neměli, ale platí, že znalost C je vhodná.

Q:Je to dokonalý a pokročilý jazyk?

A:Asi ne. Ale je velmi mocný a píše se v něm velice rychle a snadno.

Q:Kde vzít Perl?

A:Tady: <http://www.activestate.com>. Také navštivte <http://www.perl.com>.

Q:Kolik za něj?

A:Nic. Je zdarma.

Q:Instalace Perlu

A:Řiďte se instalačními poznámkami v distribuci Perlu. Všechny naše utility byly testovány na ActiveState Perlu 521-522.

Q:Jak doinstalovat některé balíky do Perlu?

A:Většinou stačí spustit program ppm dodávaný v ActiveState Perlu a napsat install balik a dál se řídit instrukcemi na obrazovce. Pro jiné případy viz nápověda Perlu.

Q:Je pro provoz iAVS nezbytný?

A:Ne. Jsou v něm jen napsány pomocné utility pro administrátory. Se stanicemi nemá nic společného.

Q:Literatura o Perlu.

A:Buď si vystačíte s anglickou dokumentací, která je dodávaná s Perlem ve formátu HTML, nebo si pořídte tzv. Camel Book.

Larry Wall, Tom Christiansen & Randal L. Schwartz: Programování v jazyce Perl, Computer Press 1997, ISBN 80-85896-95-8

Pavel Satrapa: Perl pro zelenáče, Neokortex 2000, ISBN 80-8633002-8

Q:Perl se mi líbí. Dá se s tím něco dělat?

A:Nic si z toho nedělejte, to se stává.

27 Často kladené dotazy - Unix

Q: Jak se liší postup pro práci pod Unixem?

A: Vyžaduje větší znalosti. Rozbalte soubor lindist.zip, nastavte správná práva a vlastníky pro skripty (chmod 755 *.pl *.pm). Doinstalujte potřebné balíky do perlu. Například pomocí utility CPAN (perl -MCPAN -e shell;). Závislosti jsou LWP, POSIX, Bundle::libnet, Digest::MD5, popř. další, o které si perl při zkoušení skriptů řekne. Pokud nemáte perl v /usr/bin/perl, přidejte si symlink, popř. zeditujte první řádek skriptu.

Q: Je vyžadována specifická odrůda Unixu?

A: Neměla by být, ovšem skripty byly zkoušeny pouze pod Red Hat Linux 7.0.

28 Poděkování

Při vývoji systému iAVS bylo použito následujících nástrojů a knihoven:

Kompilátory:

Microsoft Visual C++ 6.0
GNU DJGPP 2.0

Skriptové jazyky:

Perl

Knihovny:

Kompresní knihovna Zlib (Copyright (c)1995-1998 Jean-loup Gailly and Mark Adler)
Rutiny pro výpočet MD5 (Copyright (c)1999 Aladdin Enterprises, L.Peter Deutsch)
Rutiny pro binární difference (Copyright (c)2000 ALWIL Software, Jindřich Kubic, inspiro-
váno programem JENDIF (c)1997 Viliam Mlich)
Rutiny pro asymetrické šifrování (Copyright neuveden na přání autora)

Testovací servery:

Red Hat Linux 6.1
Debian 2.0
BSDI 3.0

Testovací HTTP servery:

Netscape FastTrack
Apache for Linux
Apache for Win32