Informatikai technológiák laboratórium II.

Munkaállomások távoli felügyelete

Mérési útmutató

Írták: Huszerl Gábor
Paljak Gergely
Pásztor Péter László
Tóth Dániel

Verzió: 4

2011. augusztus

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1 Bevezető

A mérési feladatok ismertetése előtt célszerű röviden áttekinteni a távoli felügyelet kialakulását, feladatait és eszközeit, valamint megismerkedni az Intel vPro platform AMT (Active Management Technology) részével. Hasonlóan rövid áttekintést adunk két további eszközről (Linux kezelése soros porton keresztül, Az AMT programozói felülete), amelyek alapszintű ismerete szintén szükséges a mérés eredményes elvégzéséhez.

1.1 Távfelügyelet

A számítógépek távoli elérése a többfelhasználós rendszerek kialakulásának egyenes következménye volt: terminálokkal oldották meg a gépek távoli, szimultán elérését. Ezeknél a rendszereknél ugyan a felhasználók távolról érték el a nagy számítógépeket, de tipikusan volt a gép mellett kezelőszemélyzet, akik reagáltak az esetleges meghibásodásokra. Az ő munkájuk hatékonyságát növelhették később azzal, hogy bizonyos üzemeltetési feladatokat ők is távolról láthattak el.

A számítógépek távoli felügyelete akkor kezdett kényelmi funkcióból égető szükségességgé válni, amikor a „server hosting" szolgáltatások elkezdtek terjedni. Ezek a szolgáltatók ugyanis nem feltétlenül vagy csak külön pénzért teszik lehetővé a 0-24 órás bejárást, közvetlen fizikai elérést. Természetesen ilyen körülmények mellett nem lehet helyszíni felügyeletet megvalósítani, esetleg olyan szolgáltatás vehető igénybe, hogy kérésre újraindítják a gépet – azonban ennél bonyolultabb feladatokat a szolgáltatók nem tudnak vállalni.

A távfelügyelet legkomolyabb felhasználása azonban nem is a szervergépek felügyelete esetén tapasztalható, hanem a kliensgépeknél. Ugyanis míg szerverből maximum pár száz van, kliens gépből egy nagyobb vállalatnál tízezernél is több lehet. Emellett minden kiszállás pénzbe és időbe kerül, illetve komoly késedelemmel jár, pedig csak hardverhiba esetén igazán indokolt a kiszállás. Napjainkban számos vállalat ad hordozható számítógépet is alkalmazottainak, ezek felügyelete külön nehézséget jelent, hiszen gyakran nem a telephelyen találhatóak meg. Ráadásul egyre több cég nem is maga felügyeli az asztali gépeit, hanem külső cégeket bíz meg ezzel a feladattal (menedzselt szolgáltatások). Ez a konstrukció azért lehet hatékony, mert kevesebb ember több gépet felügyel, viszont minden egyes kiszállás jelentősen csökkenti ezt a hatékonysági előnyt.

A legtöbb hibát maguk a felhasználók okozzák: sokszor akár jó szándék által vezérelve letörölnek fájlokat („kitakarítják" a gépet), megváltoztatják a stabil beállításokat („szétkonfigurálják" a gépet), spyware-t, trójai programot, vírust, egyéb alkalmazásokat telepítenek. Ezen hibák diagnosztizálása és kijavítása sokszor csupán pár perces feladat, azonban kiszállással együtt több órányi idejét veszik el az IT személyzetnek. A kiszállás együtt jár azzal is, hogy vagy mindig kiszállítják az összes szakértői, diagnosztikai, helyreállítási és egyéb eszközüket, vagy egy első szemrevételezés után visszamennek a telephelyükre a szükséges eszközökért. Jól működő távoli felügyelet esetén sem a teljes eszköztár folyamatos mozgatására, sem a szükséges segédeszközök nélküli munkára, sem felesleges oda-vissza rohangálásra nincsen szükség.

A leginkább fegyelmezett felhasználók mellett is időnként szükség van a meglévő alkalmazások (lehetőleg munkaidőn kívüli) frissítésére, újabb alkalmazások telepítésére. A kiszállásokra ilyenkor a telepítőmédiumok eljuttatásához és behelyezéséhez, illetve a gépek be- és kikapcsolásához, újraindításához van szükség. A távoli felügyeleti eszközök ezen holt idők csökkentésével igyekeznek növelni a hatékonyságot.

1.2 A távfelügyelet feladatai

Vállalatirányítási szempontból a távfelügyelet legmagasabb szintű feladata az IT részleg működési költségeinek csökkentése. Ez a következő alacsonyabb szintű feladatokat jelenti:

- nem hardveres eredetű hibák diagnosztizálása és elhárítása,

- a hardver és szoftver eszközök nyilvántartása,
- esetleg a hardveres hibák diagnosztizálása.

Ezen feladatokhoz szükséges az IT eszközök távoli elérése és állapotuk megváltoztatásának lehetősége. Alapvető tehát, hogy az irányítást (billentyűzet, képernyő, egér – KVM) átvegyék, esetleg eszközöket is csatlakoztathassanak hálózaton (USB eszközök, floppy, CD image stb.), illetve hogy be- és kikapcsolhassák a gépeket. A diagnosztikát teszi hatékonyabbá, ha már a rendszerindítás folyamatának üzeneteit is olvashatják távolról, illetve alacsony szintű konfigurálásokat tesz lehetővé, ha a gép BIOS-át (BIOS-ait) is tudják olvasni és módosítani. Főleg a szervergépek alaplapjai képesek a bootolás folyamán is használt karakteres képernyő tartalmát soros csatlakozáson keresztül továbbítani. Egy erre csatlakozó eszköz a nagygépes környezetből ismert felügyeleti konzol megfelelőjeként funkcionálhatna, azonban ilyen lehetőség az asztali és hordozható gépeknél ritkán áll rendelkezésre.

1.3 A távfelügyelet eszközei

Az IT rendszer üzemeltetésében a legdrágább eszköz a humán erőforrás, azaz a hozzáértő IT adminisztrátor és az ő munkaideje. Ezért minden olyan eszközt, amelynek segítségével csökkenthető az adminisztrátorok haszontalanul eltöltött ideje, be kell vetni a költségek csökkentése érdekében. Helyszíni kiszállás optimális esetben csak hardverhiba esetén engedhető meg – ezek kijavítása meglehetősen ritkán lehetséges távoli eléréssel. A IT ügyfélszolgálat (helpdesk) alkalmazása javít ugyan a helyzeten, de mivel a vonal túlsó végén egy tetszőlegesen hozzá nem értő felhasználó ülhet, sok esetben rosszabb a helyzet, mintha kimenne az IT adminisztrátor (például ha a nem szakértő felhasználó félreinformálja az adminisztrátort, vagy nem érti a helpdesk tanácsait, és ezért a szomszédja is ezzel foglalkozik – ekkor már 3 embert lefoglal a probléma), ezen felül pedig például alkalmazás átvitele a helyszínre így elég körülményes. A következőkben áttekintjük, hogy milyen szoftveres, hardveres és kombinált megoldások léteznek, valamint külön kitérünk a mérésen is használt Intel technológiára.

1.3.1 Szoftvereszközök

A távfelügyeleti szoftvereszközök illetve a nagyobb távfelügyeleti rendszerek ágensei (melyek tipikusan a felhasználó által futtatott operációs rendszerre telepített alkalmazások, szolgáltatások stb.) sok esetben megkönnyítik az IT részleg működését, hiszen ha a felhasználónak nem sikerült működésképtelenné tennie az operációs rendszert, és a megfelelő alkalmazások, szolgáltatások stb. futnak vagy elindíttathatóak, akkor a problémák nagy részét meg lehet velük oldani. Azonban ha az operációs rendszer vagy a megfelelő szerverprogram nem működik, a helyszínen kell a problémát orvosolni. Hasonló okból nem lehetséges velük sem a BIOS elérése, sem a gépek bekapcsolása fizikai elérés nélkül. Példák:

- ssh, telnet: csak szöveges üzemmódú elérés, és új konzolt nyit, nem igazi irányítás átvétel
- VNC (Virtual Network Computing), vagy RDP (Remote Desktop Protocol): igazi képernyő, irányítás, és grafikus módban is működik. Akár eszközt is csatolhat – de még mindig csak működő operációs rendszer alatt!

Mivel az ilyen eszközök a felhasználó által futtatott operációs rendszerre támaszkodnak, egy rosszindulatú felhasználó vagy a felhasználó jogosultságaival futó egyéb rosszindulatú alkalmazás által okozott károk kivédésére és helyreállítására csak korlátozottan alkalmasak.

1.3.2 Hardvereszközök

A *hardver felügyeleti eszközökre* sokáig jellemző volt, hogy drágák, és alapvetően a szerverkategóriájú gépekhez készültek. Szinte minden gyártó kínál valamilyen távfelügyeleti kártyát (pl. HP – iLO, Integrated Lights Out; IBM – RSA, Remote Supervision Adapter; Fujitsu-Siemens – iRMC, Integrated Remote Management Controller stb.), amely lehetővé teszi, hogy a gép teljes irányítását Ethernet hálózaton keresztül bonyolítsuk le. Ez teljes értékű konzol, akár grafikus felület átvitelére is alkalmas, de a sávszélesség igénye elég nagy. Az operációs rendszertől azonban nem függenek, tehát elérhető

például a BIOS. Általánosságban elmondható, hogy minden ilyen távfelügyeleti eszköz egy saját beágyazott processzorral (BMC, Baseboard Management Controller) rendelkezik, amely a gép készenléti (standby) állapotában is működik. A BMC gyakran külön hálózati interfésszel rendelkezik, és saját IP címe van. Az ilyen megoldásokat *out-of-band* menedzsmentnek nevezik, jelezvén, hogy a felügyeleti funkció valamint annak hálózati forgalma elkülönül a felügyelt operációs rendszertől és annak hálózati forgalmától.

A külső, IP alapú KVM (Keyboard, Video, Mouse) megoldásokat áruk miatt szintén inkább szerverekhez szokás használni, gyakorlatilag azonosak a tulajdonságaik az előbb említett kártyákkal, gép ki- és bekapcsolást nem támogatnak, viszont általában több géphez is használhatóak, ami kifejezetten hasznos. Igen nagy a sávszélesség igényük, hiszen bemenetükön csak analóg adatokat kapnak videojelként, azt digitalizálják, és küldik tovább a hálózaton.

Az alaplapi chip készletbe integrált hardveres KVM (pl. Intel KVM Remote Control) már felső kategóriás asztaligép alaplapokon is elérhető, BMC-szerű külön vezérlőjének köszönhetően a fenti megoldások előnyeit próbálja kombinálni: nem külső eszköz, nem kell külön megvenni, független a felügyelt gép operációs rendszerének működőképességétől, a bootolási folyamat kezdetétől teljes megfigyelhetőséget és beavatkozást biztosít.

Hardvereszközként kategorizálható a már jó ideje használható, azonban kényelmetlensége miatt ritkán használt Wake-on-LAN megoldás. Ennek lényege, hogy a számítógép egy külön Ethernet keret hatására bekapcsol. Ez azonban elég megbízhatatlan megoldás, hiszen egyoldalú kommunikáció, ráadásul ahhoz, hogy más felügyeleti eszközt használjunk, működő operációs rendszer kell. Ráadásul a praktikus alkalmazást jelentősen korlátozza az is, hogy a csomagot küldő gépnek azonos hálózati szegmensben kell lennie a fogadó géppel (ez igazi távfelügyeletnél csak bonyolultan konfigurált virtuális hálózatokkal oldható meg). Ezek miatt tehát bár segítheti a távfelügyeletet, segítségével bizonyos feladatok – például működő gépek éjjeli frissítése, vírusirtása – automatizálhatóak, de csak kényelmetlenül, bizonytalanul, és autentikáció nélkül.

A hardveres távfelügyelet protokolljai

A különböző gyártók BMC megoldásainak szabványos kezelésére elterjedt protokoll az **IPMI** (Integrated Platform Management Interface). A fent felsorolt gyártó-specifikus megoldások az IPMI különböző verzióit általában támogatják. Újabban kezd megjelenni a Web Services technológiákra építő WS-Management is távfelügyeleti alkalmazásokban. A WS-Man tetszőleges CIM (Common Information Model) osztályokhoz nyújt hozzáférést, a szerverek egységes szabványos távfelügyeletéhez szükség volt az osztálystruktúra szabványosítására is. Erre jelenleg a Distributed Management Task Force (DMTF) készített ajánlásokat:

- Systems Management Architecture for Server Hardware (**SMASH**) – főleg szerverkategóriájú eszközök kezelésére
- Desktop and mobile Architecture for System Hardware (**DASH**) – főleg asztali és hordozható eszközökhöz

Mindkét ajánlásról elmondható, hogy a központi elemeik a CIM osztály *profilok*, melyek egy-egy részfeladatot fednek le. Például külön profil van a boot opciók kezelésére, hardver szenzorok elérésére, KVM átirányítás vezérlésére stb. A DASH vagy SMASH megfelelőséghez a gyártóknak néhány alap profilt (pl. jogosultságkezelés) kötelezően meg kell valósítaniuk, a többit (pl. akkumulátor kezelése) opcionálisan valósíthatják meg, így a támogatott CIM osztályok halmaza eltérő lehet.

A SMASH profilokat már számos hardvergyártó használja (pl. IBM BladeCenter Advanced Management Module, DELL Blade Management), sőt virtuális szerverek kezelésére a VMware ESX Server újabb változataiba is bekerült. DASH profilokat jelenleg az Intel AMT technológiája valósít meg.

1.3.3 Intel vPro

Az Intel vPro egy alaplapi technológiákat magában foglaló platform, amely nagyvállalati felhasználók igényeit hivatott kielégíteni. Nagyvállalati környezetben nem ritka a több ezer, tízezer asztali számítógép, amelyek a nem feltétlenül erősen szakképzett felhasználókkal együtt igen komoly

ráfordítást igényelnek a nagyvállalati IT részlegtől, és ez akár igen komoly működési költséget is jelenthet. Ezt a költséget hivatott minimalizálni ez a platform, amelynek két eleme van, ezek a Virtualization Technology (VT) és az Active Management Technology (AMT).

Intel VT

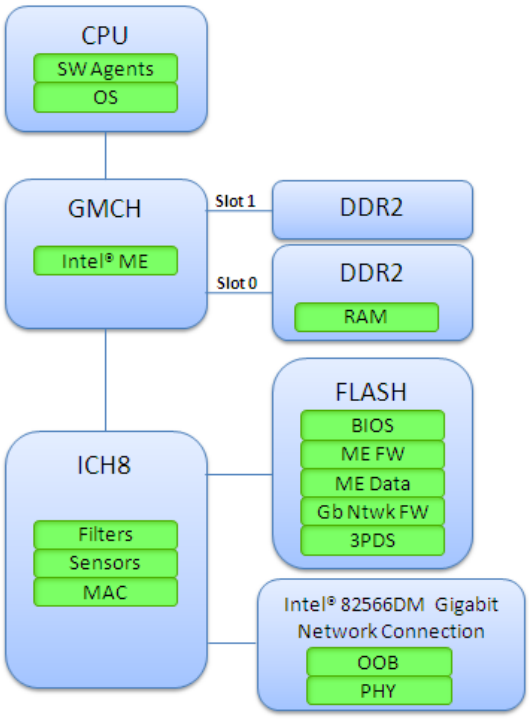
Az Intel VT egy a processzorba épített virtualizációt támogató utasításkészlet, amelynek segítségével a számítógépen lehetséges egyszerre egynél több, nem módosított operációs rendszer futtatása. Ezt kihasználhatjuk úgy, ahogyan a VMware vagy Xen virtualizációs szoftverek is teszik, de a vPro platformban más szerepet szántak neki. Az eredeti elképzelés szerint a felhasználó által látott/használt „fő" operációs rendszer mellett egy kicsi, láthatatlan, „biztonsági" rendszer is futna, amely korlátozhatja, felügyelheti a gép, az operációs rendszer műveleteit, csökkentve a vírusok, támadások kockázatát vagy – akár pénzben is kifejezhető – hatásait. Ez a megoldás nem igazán terjedt el, az Intel VT-vel jelen mérés során nem foglalkozunk.

Intel AMT

Az Intel AMT egy olyan, alaplapi chipsetbe integrált (tehát a processzortól és az Intel VT technológiától gyakorlatilag független, ld. 1. ábra) hardveres felügyeleti platform, amely a számítógépen futó alkalmazásoktól nagymértékben független („out-of-band") működésre képes.

Az AMT fő hardverkomponensei[1]:

- Az alaplapi chipset északi hídjában (MCH) található egy beágyazott CPU. Ez az AMT vezérlő, a neve *Management Engine* (ME). A beágyazott processzor a számítógép kikapcsolt, de áram alá helyezett (standby) állapotában is működik. Notebookok esetén konfigurálható, hogy akkumulátoros üzemben is működjön-e (akku töltésének kímélése céljából csak Wake-on-LAN funkcióval aktiválható) vagy csak tápegységre kapcsolt állapotban.
- A beágyazott CPU programját és adatait tárolja a déli hídhoz (ICH) kapcsolt firmware hub (általában egy flash memória modul). A Management Engine egy saját firmware-rel rendelkezik, ami nem azonos a gép BIOS-ával.
- A chipset déli hídja tartalmaz néhány módosítást az alapmodellhez képest: például a hardver szenzorok kezeléséért az AMT felelős, valamint tartalmaz egy beépített hardveres tűzfalat is, amit szintén az AMT vezérel. Itt kapott helyet a floppy és CD meghajtó átirányításhoz szükséges periféria is.
- A hálózati vezérlő is speciális, egy fizikai kapcsolaton két független hálózati interfészt kezel, az egyik az AMT kommunikációjáért felelős, a másikat használja a gép operációs rendszere. Az AMT portja a gép standby állapotában is működik, és képes forgalmazni. Az AMT notebookokba szerelt változatában a vezeték-nélküli hálózati vezérlő is hasonló képességekkel rendelkezik.



1. ábra: Az Intel AMT hardver komponensei

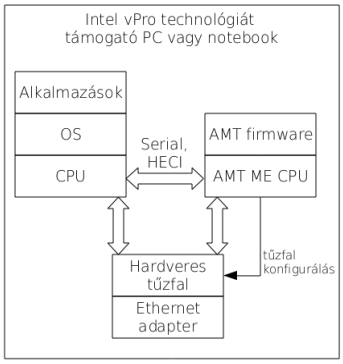
ICH ─ I/O Controller Hub (déli híd);
MCH ─ Memory Controller Hub (északi híd);
ME ─ Management Engine

[1]Az alábbi leírás az AMT felépítésének egy egyszerűsített változatát ismerteti. Az AMT újabb és újabb változatai ezt a felépítést részben egyre újabb kisebb komponensekkel bővítik, másrészt az elnevezések is időről időre változnak, azonban az alapvető felépítés marad.

Ezeket a hardverkomponenseket az Intel Q szériás chipkészlettel (Q965, Q35, Q45, Q57) szerelt alaplapokon találhatjuk meg. Fontos hangsúlyozni, hogy ezek alapvetően közönséges asztali számítógépekbe szánt alaplapok, a „márkagépek" szállítóin (Lenovo, Dell, HP) kívül számos közismert alaplapgyártó (Intel, ASUS, Gigabyte) kínálatában is megtalálhatók. Az AMT nélküli változataikhoz képest minimális felárral kaphatóak, és tényleg megtalálhatóak kiskereskedelmi forgalomban. Notebookok esetén jellemzően a „business" kategóriás típusokat szerelik fel AMT-vel.

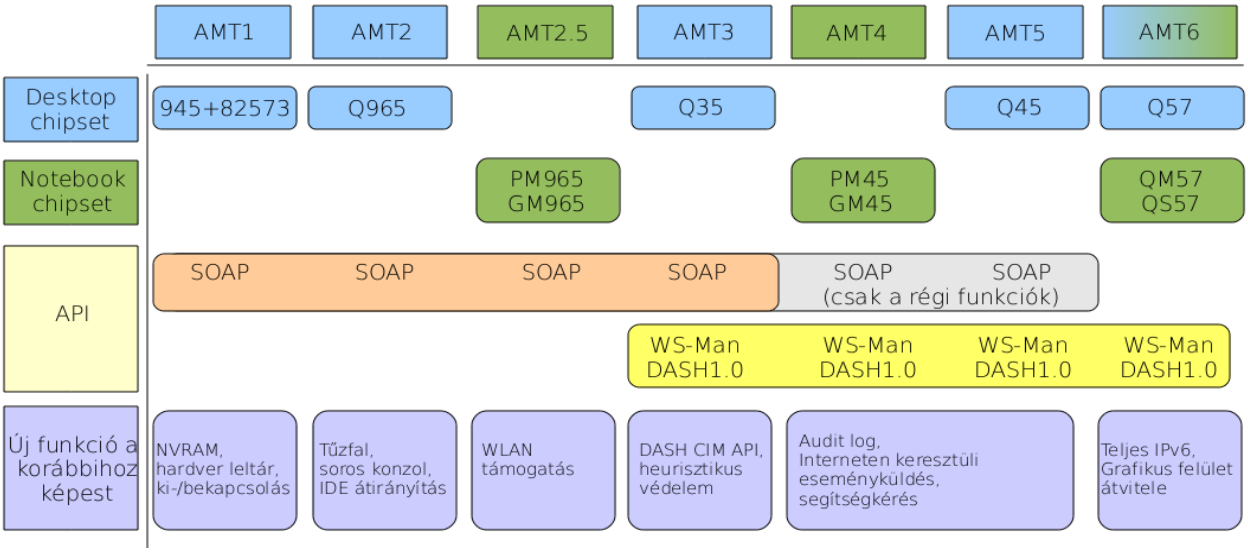
Az AMT vezérlő ugyan fizikailag ugyanazt a hálózati csatolót használja, mint a számítógép többi része, ennek ellenére akár saját IP-címe is lehet, vagy használhat a gép operációs rendszerével közös IP címet is. Hogyan lehetséges ez? A chipset beépített hardveres tűzfala TCP portok szerint képes szétválogatni a forgalmat, így az AMT vezérlésére szánt portokra (TCP 16992-16994) érkező csomagok az ME-hez, a többi forgalom a gép fő CPU-ján futó operációs rendszerhez kerül.

A gép fő processzora kétféle periférián keresztül képes az AMT beágyazott CPU-val kommunikálni: egy belső soros porton valamint az ún. HECI (Host Embedded Controller Interface) eszközön át (ld. 2. ábra).



2. ábra Az Intel AMT logikai felépítése

Az AMT verziónként eltérő képességekkel rendelkezik (ld. 3. ábra). Általánosságban a fő (major) verziószám a hardver képességeit határozza meg, tehát fő verzió firmware frissítéssel nem léptethető. Az al- (minor) verziószám szoftveres frissítéseket takar, firmware frissítéssel ez léptethető. Ez alól a séma alól kivétel az AMT 2.5, ami egy különálló fő verziónak számít.[2]



3. ábra: Az Intel AMT verziók áttekintése

A mérés szempontjából fontos, hogy a laborgépek fele AMT2-es verziójú (a másik fele AMT6-os verziójú), ezért nem támogatja még a WS-Man feletti vezérlést (az ehhez szükséges firmware egyszerűen nem fér el a korábbi változatoknál alkalmazott flash memóriában). Létezik egy WS-Mananagement Translator szoftver komponens, ami képes a WS-Man hívásokat a régebbi saját függvényeket használó SOAP Web-Service API-ra lefordítani. A régebbi API-hoz elérhetőek a WSDL leírások, így ehhez is lehet generálni kliensprogramot.

[2]2011 őszén már létezik az AMT 8.0 verziója is, de ilyen összehasonlító ábra nem készült újabb. Érdekes információk találhatóak a http://en.wikipedia.org/wiki/Intel_AMT_versions oldalon is. Sajnos a publikus dokumentumok nem tartanak lépést a technológia fejlődésével.

Az Intel AMT két üzemmódban használható (ezek az elnevezések is változtak, de a lényeg maradt, és az újabb üzemmódok is csak ezek alváltozatai):

- A *Small business* üzemmódot alapvetően a kis és közepes méretű vállalatokra szabták, amelyeknek nem áll rendelkezésére komoly kiszolgáló infrastruktúra, de cserébe a belső hálózatuk tipikusan zárt, nem megy keresztül publikus (és ezért alapvetően megbízhatatlan) hálózati szegmenseken. Itt az egyszerűbb üzemeltetés érdekében lemondunk az automatikus konfigurálásról, így a gépeket először kézzel (vagy egy „pen drive”-val) be kell állítani.
- Az *Enterprise* üzemmód ezzel szemben a nagyvállalati szegmenst célozza meg, erős TLS titkosítással, AMT konfigurációs szerverekkel. Az AMT konfigurációs szerverek például az AMT beállítások automatikus, nagybani terítését végzik. Ehhez a mérés során sem fog rendelkezésre állni a szükséges infrastruktúra.

Az AMT használata számos előnnyel bír a szokványos eszközökhöz képest, de önmagában természetesen nem elégséges a teljes IT felügyelethez. Alapvetően azt a helyzetet hivatott megoldani, amikor az operációs rendszer üzemképtelen, vagy a hálózati kapcsolatot valamilyen okból blokkolni kellett, de a hardver hibátlan. Ekkor bár a probléma szoftveres, mégsem lehetséges a hiba távoli megoldása, mivel a felügyeleti rendszer működéséhez kellene az operációs rendszer, és legalább részben engedélyezni kellene a hálózati forgalmat. De az AMT akár olyan helyzetben is nagy segítséget nyújthat, amikor az operációs rendszer ugyan működik, de például a felügyeleti rendszer ágensei vagy a vírusirtó már nem. (Az ilyen ágensek és kritikus eszközök működőképességét a ME bizonyos fokig ellenőrizni is tudja, és szükség esetén riasztással vagy az operációs rendszer hálózatról való leválasztásával is reagálni tud.)

Mivel az AMT működéséhez nem szükséges operációs rendszer, így ha a hardver hibátlan, segítségével egy jól felépített felügyeleti rendszerben minden szoftverprobléma orvosolható távolról. Megfelelő körülmények között megoldható még akár az operációs rendszer teljes újratelepítése vagy a BIOS beállítása is úgy, hogy a gép közben akár egy másik kontinensen egy teljesen üres és lezárt irodában is lehet.

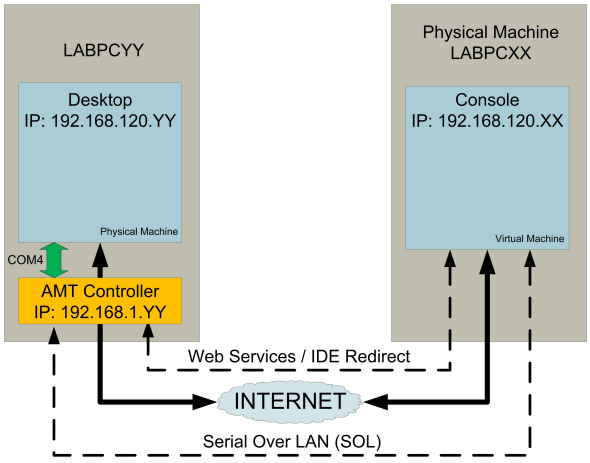
Az AMT segítségével automatikusan, a gépek kikapcsolt állapotában végezhető hardver leltár, illetve a beépített flash memóriás tároló adatai is elérhetőek. (Alkalmazási példa: aznap bejelentkezett felhasználók listája.) A gépek az AMT segítségével távolról bármikor be- és kikapcsolhatóak – ez hardveres, tehát nem ACPI (Advanced Configuration and Power Interface) kikapcsolást jelent. Így rengeteg feladat hatékonyan elvégezhető, de ami még fontosabb, a technológiát a nagyvállalati központi IT felügyeleti rendszerbe integrálva akár automatizálható is. (Pl. éjszakai alkalmazás- vagy rendszerfrissítés, vírusirtó futtatás, lemezek töredezettség-mentesítése stb.)

1.4 Az Intel AMT részei

Röviden ismertetjük az AMT platform néhány komponensét. Viszonylag „kézzel fogható” komponensként tekinthetünk a felügyeleti konzol és a felügyelt gép között kialakított virtuális soros csatornára és a felügyelt gépbe épített hardveres tűzfalra. Inkább szolgáltatás komponens a hálózati forgalom átirányítás, a beépített rendszervédelem és az ágensek jelenlét-ellenőrzése.

Virtuális soros csatorna (SOL)

A Serial over LAN (SOL), ahogy a neve is mutatja, hálózaton keresztülbújtatott soros kommunikációt takar (ezért a gép hálózati csatolóján kisajátítja a 16994 TCP portot). Ez a virtuális soros kapcsolat (ld. 4. ábra) a felügyeleti konzol (az adminisztrátor gépe) és az ME között épül ki, a felügyelt gép többi része (BIOS, operációs rendszer stb.) számára szabványos soros csatlakozásként jelenik meg.



4. ábra: A felügyelt gép és a felügyeleti konzol hálózati kapcsolata

Ezen a soros kapcsolaton keresztül a régi nagygépes rendszerek terminálos rendszeréhez hasonló, de Ethernet hálózaton folytatott kommunikáció építhető ki, aminek segítségével az AMT pl. a konzol információit (billentyűzet, karakteres módú megjelenítés) átviheti. Grafikus módú képernyő adatai soros vonalon keresztül nem vihetőek át, ezt VNC protokoll segítségével valósították meg az AMT6.0 verziótól. Korábbi változatokban egyáltalán nincs grafikus konzol támogatás, így a mérési környezetben sem.

A SOL segítségével távolról lehetséges a rendszerindítás megfigyelése, közbeavatkozás, rendszerindítás más médiumról, például a felügyeleti konzolba behelyezett közönséges floppy lemezről vagy CD-ről, esetleg ezek fájlba mentett képéről (*.img, *.iso). Megfelelően összeállított karbantartó eszközökkel (pl. karakteres módú: FreeDOS bootfloppy, nem grafikusan bootoló Linux CD, VNC-vel rendelkező grafikus: BartPE, Windows PE, Knoppix, 911cd.net) bármilyen nem-hardveres feladat elvégezhető.

A kapcsolat sávszélessége tipikusan a régi modemes időket idézi, de ez egyrészt a legtöbb felügyeleti feladathoz elegendő, másrészt akár a nagyvárosi vagy távolsági utazási időkkel is versenyképes. (Ráadásul kézenfekvővé teszi a régi modemes és terminálos kommunikációnál bevált protokollok újrafelhasználását.)

Hardveres tűzfal (Firewall)

A hálózati csatolóval való szoros kapcsolat miatt logikus volt, hogy a hálózati forgalmat szabályozhatjuk az AMT segítségével, így egy operációsrendszer-független, hardveres tűzfal áll rendelkezésre minden AMT-t támogató számítógépen. Triviális felhasználási módja a bemenő kapcsolatok szűrése, de akár a kimenő kapcsolatokat is lehetséges korlátozni. Az AMT operációs rendszer szintű meghajtóprogramja segítségével is vezérelhető, például letiltható a teljes hálózati kommunikáció, ha a felhasználó – vagy egy rosszindulatú vírus – kiiktatja az antivírus programot. Amennyiben a gép teljes karanténba került, az AMT még természetesen elérhető, mivel az „kívül” van a tűzfalon. Emiatt persze támadható is, erre azonban a hozzáférésnél Enterprise módban az SSL alapú kommunikáció illetve a Kerberos alapú autentikáció nyújt védelmet.

Hálózati forgalom átirányítása (Port Redirection)

Az alapvetően működőképes, de a hálózatról valamiért leválasztott felügyelt gépen futó egyes hálózati szolgáltatásokat elérhetővé tehetünk a hálózati forgalom átirányításával. Például a karanténba került gép 80-as portján az amúgy működőképes web szerver sem kap kéréseket. Az átirányítással azonban elérhető, hogy a felügyeleti konzol egy kiválasztott portjára (pl. 80, 88 vagy 8080) érkező forgalmat a konzol a SOL kapcsolaton átküldje az AMT vezérlőnek, mert ezt a kapcsolatot a karantén nem érinti. A vezérlő a csomagokat átadja az operációs rendszer hálózatkezelő részének, mintha azok a 80-as portra érkeztek volna. A válasz forgalom ugyanezen az úton ki is tud jutni. Ez az átirányítás akkor is működhet, ha az operációs rendszer más okból nem látja a hálózatot (pl. félrekonfigurálás).

Beépített rendszervédelem (System Defense)

A hagyományos védelmi rendszerek (pl. Intrusion Detection Systems, Virus Detection) egyik komoly problémája a behatolás érzékelése és az ellencsapás között szükségszerűen eltelő idő. Az AMT beépített rendszervédelme segítségével hálózati védelmi rendtartásokat (policies) írhatunk elő és kényszeríthetünk ki, amelyek általunk definiált szűrőknek (filters) a felügyelt gép hálózati csatlakozásán kimenő és bejövő forgalomra való illeszkedésekor általunk meghatározott akciókat válthatnak ki.

A rendtartásokat, szűrőket és akciókat a felügyeleti konzolon adhatjuk meg, de azután az AMT vezérlő futtatja őket. Mivel a vezérlő a processzor és a hálózati csatlakozás között helyezkedik el, ezért az operációs rendszertől és az azon futó programoktól függetlenül minden hálózati forgalmat lát, és a szűrő által megfogalmazott mintára való illeszkedés esetén tetszőleges forgalmat késlekedés nélkül jegyezni, számolni vagy akár blokkolni képes. Az AMT vezérlő (és minden, ami rajta keresztül elérhető) ugyanakkor teljes blokkolás mellett is távolról elérhető marad.

A beépített rendszervédelem jelenlegi neve *System Defense*, a dokumentációkban és az API-ban helyenként felbukkanó CB rövidítés a korábbi *Circuit Braker* elnevezésre utal.

Ágensek jelenlét-ellenőrzése (Agent Presence)

A hagyományos védelmi rendszerek másik komoly problémája, hogy ha a felhasználó vagy az ő nevében „valaki" más kiiktatja a védelmi rendszer egy részét, akkor ezt mikor és miből vehetjük észre. Az AMT képes folyamatosan ellenőrizni különböző alkalmazás ágensek (vírusirtó, tűzfal stb.) jelenlétét. Az AMT vezérlő periodikusan szólítgatja (ping) az ágenseket, amelyek normál esetben ezt mindig nyugtázzák (heartbeat). Az ágens természetesen nem válaszol, ha nem fut, vagy az általa figyelt alkalmazás komponens nem fut. A válasz elmaradása az AMT vezérlőben hálózat védelmi rendtartásokat aktiválhat, ami a fentebb leírt tűzfalszerű akciókat indíthat. Ennek a módszernek a reakcióideje a szólítgatás periódusidejétől függően pár másodperc nagyságrendű.

1.4.1 Intel AMT Manageability Developer Tool Kit

Az Intel AMT tehát tartalmaz néhány alaplapi chipset kiegészítést, és ezek igen hasznos szolgáltatások nyújtására képesek, de az elérhetővé tett funkcionalitások használatához további komponensekre van szükség. Ezeket a felügyeleti rendszert kialakító cégnek vagy szervezetnek kell elkészítenie, tipikusan a felügyeleti eszközök gyártója építi bele termékébe. A technológia kipróbálására azonban az Intel készített egy demó alkalmazást, amely „Intel AMT Manageability Developer Tool Kit" (Intel AMT DTK vagy Intel AMT Manageability DTK) néven érhető el. Ennek használatát több publikus cikk, videó és fórum is részletezi. Ez az alkalmazás éles ipari célokra nem alkalmas, de egyrészt jól szemlélteti az AMT képességeit, másrészt publikus kódjának tanulmányozásával megismerhető a fejlesztők elképzelése az alkalmazásáról. Az alábbiakban a DTK néhány komponensét mutatjuk be, a fejlesztőkészlet azonban további (részben érdemi, részben a demonstrációt segítő) komponenst is tartalmaz.

Host Embedded Controller Interface (HECI) meghajtó

A felügyelt gépen futó felhasználói operációs rendszer (illetve annak az alább leírt szolgáltatása) és az AMT vezérlő közötti kapcsolattartáshoz szükség van egy meghajtó programra, ezt a feladatot a DTK-ban a HECI (Host Embedded Controller Interface) driver látja el. Ennek telepítése nélkül az AMT-nek kizárólag azok a funkciói működőképesek, amelyekhez nincsen szükség az operációs rendszer és az azon futó szolgáltatások, alkalmazások közreműködésére. A soros port a HECI meghajtó nélkül is működőképes.

Manageability Outpost Tool Agent szolgáltatás

Szintén a felügyelt gépen kell futnia egy olyan operációs rendszer szolgáltatásnak, amely kapcsolatot tud tartani egyrészt a fenti meghajtó programon (pl. HECI) keresztül az AMT vezérlővel és azon keresztül a felügyeleti konzollal, másrészt pedig az operációs rendszeren futó alkalmazásokkal és szolgáltatásokkal. A DTK ezen komponensét nevezik Manageability Outpost Tool Agent szolgáltatásnak.

Outpost ágens

A fenti szolgáltatást a DTK-ban több alkalmazás is használja. Ezek egyikének feladata, hogy egy olyan karakteres konzolt hozzon létre az (akár amúgy grafikus felületet használó) operációs rendszeren, amelyet az AMT vezérlő és a SOL segítségével a felügyeleti konzolról is kezelni lehet. Ezen a konzolon keresztül az IT adminisztrátor a távolból tud a működő rendszerre fájlokat fel- és letölteni, processzeket lekérdezni, indítani és leállítani, egyszerűbb parancssoros utasításokat kiadni, és azok eredményét olvasni. Ilyen módon ez az ágens az IT felügyelet előretolt állása (outpost) az operációs rendszeren.

AMT vezérlő

A DTK egyik fontos eleme az AMT Commander névre hallgató felügyelői alkalmazás, amely az IT adminisztrátor konzolján futva elérhetővé teszi az AMT felügyeleti funkcióit. Jelszavas azonosítással az adminisztrátor elérheti a felügyelendő gépeket:

- megtekintheti a gép hardver leltárát,
- elindíthatja, leállíthatja, újraindíthatja a felügyelt gépet,
- átirányíthatja a lemez meghajtóit (ez bootoláskor igen hasznos lehet),
- csatlakozhat az Outpost ágenshez és az általa nyújtott konzolhoz valamint egyéb segédeszközökhöz,
- védelmi rendtartásokat és felügyelő ágenseket definiálhat, aktiválhat,
- az esetleg a hálózatról leválasztott felügyelt gép hálózati portjait elérhetővé teheti a saját gépén (és a SOL kapcsolaton) keresztül,
- az adminisztrátorok számára nyújtott további funkciókat elérhet, kipróbálhat.

1.5 Linux kezelése soros porton keresztül

Ebben az alfejezetben egy a távfelügyeleti technológiákhoz és az Intel AMT megoldáshoz kevésbé szorosan kapcsolódó technikáról lesz szó, amely azonban a mérési feladatok végrehajtásához nélkülözhetetlen.

A klasszikus UNIX rendszerekhez hasonlóan a Linux is képes a konzol ki és bemenetét a képernyő és billentyűzet helyett átirányítani soros portra. A működés megértése szempontjából fontos megemlíteni az elsődleges konzol fogalmát, amit a kernel használ az üzenetei megjelenítésére rendszerindítás közben és – ha később át nem konfigurálja például egy naplógyűjtő démon – további üzeneteit is ide fogja küldeni. Ez az eszköz a `/dev/console` fájlnéven érhető el. A kernel elsődleges konzolját a rendszerindításkor a kernelnek átadott parancssori paraméterrel adhatjuk meg. A konzol átirányítása a rendszer első soros portjára a következő paraméterrel tehető meg: `console=ttyS0,115200` ahol a `ttyS0` jelenti az első soros portot (más elnevezési rendszerben ez lenne a COM1), a `115200` pedig az átviteli bitráta. Ahhoz, hogy ténylegesen megkapjuk a kernel üzeneteit, figyelnünk kell arra, hogy más parancssori paraméterek ne tiltsák ezt le (`quiet`, `splash`).

A kernel üzeneteinek átirányítása más eszközre nem vonja magával a login prompt átirányítását is. A felhasználó parancssori beléptetéséért felelős login programot a `getty` nevezetű program indítja el, melynek feladata egy konzol üzemmódjának beállítása, egy előre meghatározott üzenet megjelenítése (`/etc/issue` fájl, általában az adott rendszer nevét és verziószámát tartalmazza). Amikor a felhasználó elkezd belépni, a vezérlés átkerül a login programhoz, amely ellenőrzi a jelszót, sikeres belépés esetén kiír egy üdvözlő üzenetet (`/etc/motd` – message of the day), és elindítja a parancsértelmezőt. Tehát ahhoz, hogy a soros konzolon login promptot kapjunk, a `getty` programot kell elindítani a következő paraméterezéssel: `sudo getty /dev/ttyS1 115200`. A `sudo`-ra azért van szükség, mert az indítandó login folyamat már a belépő felhasználó nevében kell, hogy fusson, ehhez pedig a `getty`-nak rendszergazdai jogosultságokkal kell futnia. A gép indításakor automatikusan elinduló getty folyamatokat általában a `/etc/inittab` konfigurációs fájl írja le, ennek részleteibe a mostani mérés során nem megyünk.

Fontos megjegyezni, hogy a telnet, ssh és hasonló távoli szöveges elérést biztosító protokollokhoz látszólag nagyon hasonló az AMT által biztosított soros elérés. Azonban az előbbiek az operációs

rendszer hálózati alrendszerére (és annak helyes beállításaira) támaszkodnak, ellenben az AMT ettől teljesen függetlenül működik, így nem fordulhat elő, hogy egy távolról végzett hibás beállítással „kizárjuk" magunkat a felügyelt gépről (tipikusan tűzfal konfigurálásakor szokott előfordulni). Hasonlóképpen az AMT hardveres tűzfalával szükség esetén az operációs rendszert elvághatjuk a hálózattól, így saját magunkat a operációs rendszer felett futó összes távoli elérést biztosító szoftvertől is, viszont az AMT soros konzolja ilyenkor is működképes marad.

1.6 Az AMT programozói felülete

1.6.1 Intel AMT SOAP Web Services

Az AMT 2.0-s laborgépek csak az Intel saját metódusait használó SOAP interfészt támogatják. Az Intel AMT SDK-ban az `Intel(R)_AMT_SDK_6.0_Hot_Fix_2450/DOCS/WSDL` könyvtár alatt megtalálhatóak a webszolgáltatás leíró (WSDL) fájlok, amelyekből kliensprogram generálható. A számos WDSL fájl különböző funkciók interfészeit írja le:

AgentWatchdogLocalInterface.wsdl, AgentWatchdogRemoteInterface.wsdl	az operációs rendszeren futó ágens állapotának megfigyelése
AuditLogInterface.wsdl	az elvégzett műveletek naplózása
CircuitBreakerInterface.wsdl	a hardveres tűzfal beállítása
EndpointAccessControlAdminInterface.wsdl, EndpointAccessControlInterface.wsdl	hálózati hozzáférési protokollok (Cisco NAC, Microsoft NAP) támogatása
EventManagerInterface.wsdl, UserNotificationInterface.wsdl	riasztás egyes eseményekről
FirmwareUpdateInterface.wsdl	BIOS és AMT firmware távoli frissítése
GeneralInfoInterface.wsdl	általános beállítások: hoszt név, AMT üzemódja, engedélyezett funkciók
HardwareAssetInterface.wsdl	hardver leltár
NetworkAdministrationInterface.wsdl	az ME hálózati beállításai
RedirectionInterface.wsdl	IDE és floppy átirányítása
RemoteAccessAdminInterface.wsdl	hozzáférés házirend alapú szabályozása (AMT4.0-tól)
RemoteControlInterface.wsdl	üzemállapot váltása, indítás, leállítás, BIOS-ba belépés
SecurityAdministrationInterface.wsdl	felhasználói hozzáférési listák (AMT 4.0 előtt)
StorageAdministrationInterface.wsdl, StorageInterface.wsdl	alaplap flash tárolójában elhelyezhető saját adatblokkok (3DPS – 3rd Party Storage) kezelése
UserAccessControlInterface.wsdl	hozzáférési listák (AMT 4.0 után)
WirelessConfigurationInterface.wsdl	vezeték nélküli hálózati interfész beállításai

Néhány műveletnél (pl. remoteControl) szükség van az *ianaOEMNumber* megadására, ami az Intelnél 343.

1.6.2 WS-Man translator

Az AMT 3.1 előtti verziói számára elérhető a WS-Man Translator, amely – a későbbi AMT verziók által már támogatott – DASH profiloknak megfelelő CIM osztályokat teszi elérhetővé, ám a műveletek elvégzéséhez valójában a régi SOAP interfészt használja. A Translator egy windows szolgáltatás, mely értelemszerűen nem a felügyelt távoli gépen fut, hanem a felügyelő gépen. Emiatt a WS-Man klienssel (pl. `winrm`) nem közvetlenül a távoli gépre kell kapcsolódni, hanem a Translatort futtató gépre, a szolgáltatás URI-ban kell megadni a távoli gép címét, ahova a Translatornak kell továbbcsatlakoznia. Például, ha a felügyelt gép IP címe 10.40.120.130, és a Translator a helyi gépen fut, akkor az URI: `http://localhost/wstrans/pro/eoi20/10.40.120.130/wsman`. Az `eoi20` azt jelenti, hogy a felügyelt gép AMT verziója 2.0-s. A translatort a `wstranscfg` programmal konfigurálhatjuk.

Az Intel AMT SDK tartalmazza az összes AMT verzió által támogatott osztályok referenciáját egy offline böngészhető webes dokumentáció formájában:

Intel(R)_AMT_SDK_6.0_Hot_Fix_2450/DOCS/Implementation and Reference Guide/default.htm

Alapvetően háromféle osztály van:

- a CIM kezdetűek a szabványos DASH sémából jönnek,
- az AMT kezdetűek az Intel gyártóspecifikus saját osztályai,
- az IPS kezdetűek szintén gyártóspecifikus osztályok, de csak AMT 6.0-tól érhetőek el.

WS-Man osztályok elérésénél nemcsak az osztály nevére, hanem egy teljes elérési utat tartalmazó URI-ra is szükség van:

- AMT osztályoknál: http://intel.com/wbem/wscim/1/amt-schema/1/
- CIM osztályoknál: http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/

A dokumentációban érdemes még megfigyelni, hogy az Intel AMT Features fejezet alatt minden képességhez találhatóak használati eset példák, melyek leírják, hogy néhány tipikus művelethez mely osztályok mely attribútumait és metódusait kell felhasználni.

Objektum referenciák kezelése

Sok metódus meghívásakor szükség lehet az egy-egy osztály példányára mutató referenciák paraméterként való átadására. Ezt sajnos a `winrm` csak nagyon nehézkesen teszi lehetővé, külön XML fájlban kell összeállítani az elküldendő kérést az összes paraméterrel. Ráadásul a `winrm` sajnos egyszerű osztályokra nem is tud referenciát adni, de pl. asszociációs osztályok végpontjainak referenciáit képes XML szelektor kifejezésként visszaadni.

Egy példa a számítógép bekapcsolására az API dokumentáció példáját követve:

1. Keressük meg a felügyelt hosztot leíró *CIM_ComputerSystem* példányt.

   Fontos a *basic* autentikáció (a Translator beállítható lenne ennél erősebbre is) és az *utf-8*-as karakterkódolás beállítása.

   ```
   winrm enumerate http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
   -remote:http://localhost/wstrans/pro/eoi20/10.40.120.130/wsman -a:Basic -u:admin
   -p:AMTje1szo% -encoding:utf-8
   ```

   A fenti parancsra például az alábbi választ kaphatjuk:

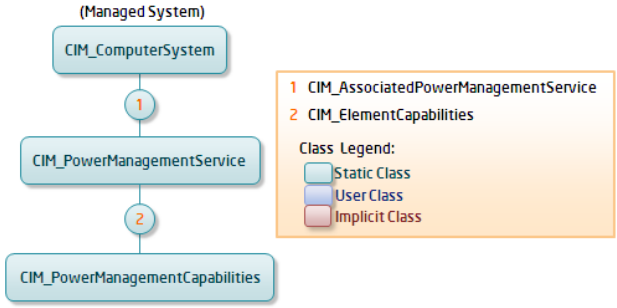
   ```
   CIM_ComputerSystem
       Name = Intel(r) AMT
       CreationClassName = CIM_ComputerSystem
       ElementName = Intel(r) AMT Subsystem
       OperationalStatus = 0
       HealthState = 5
       EnabledState = 5
       RequestedState = 12
       EnabledDefault = 5
       NameFormat = Other
       Dedicated = 14
   CIM_ComputerSystem
       Name = ManagedSystem
   ```

```
CreationClassName = CIM_ComputerSystem
ElementName = Managed System
OperationalStatus = 0
HealthState = 5
EnabledState = 3
RequestedState = 10
EnabledDefault = 5
NameFormat = Other
Dedicated = 0
```

Két példányt kaptunk, ami nem meglepő, hiszen az egyik a felügyelt hoszt gép, a másik pedig az AMT beágyazott számítógép rendszere. Nekünk a „Managed System” nevű példány kell.

2. Szerezzük meg a hozzá kapcsolódó *CIM_PowerManagementService* objektumot.

Itt kihasználjuk, hogy az osztálydiagramon a *CIM_ComputerSystem* és *CIM_PowerManagementService* között van a *CIM_AssociatedPowerManagementService* asszociáció (ld. 5. ábra), így ezt kérdezzük le, mégpedig a végpontokra referenciákkal (EPR – EndPoint Reference). Ehhez a `format:pretty`-vel ki kell választani az XML kimeneti formátumot és a `ReturnType:EPR`-rel kérni referenciát.



5. ábra: CIM osztálydiagram részlet

```
winrm enumerate http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_AssociatedPowerManagementService -remote:http://localhost/wstrans/pro/eoi20/10.40.120.130/wsman
-a:Basic -u:admin -p:AMTje1szo% -encoding:utf-8 -format:pretty -ReturnType:EPR
```

```
<wsman:Results xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman/results">
<p:CIM_AssociatedPowerManagementService
xml:lang=""
xmlns:p="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AssociatedPowerManagementService"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
    <p:PowerState>8</p:PowerState>
    <p:RequestedPowerState>8</p:RequestedPowerState>
    <p:ServiceProvided>
        <a:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</a:Address>
        <a:ReferenceParameters>
            <w:ResourceURI>
                 http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PowerManagementService
            </w:ResourceURI>
            <w:SelectorSet>
               <w:Selector Name="Name">Intel(r) AMT Power Management Service</w:Selector>
                <w:Selector Name="CreationClassName">CIM_PowerManagementService</w:Selector>
                <w:Selector Name="SystemName">Intel(r) AMT</w:Selector>
                <w:Selector Name="SystemCreationClassName">CIM_ComputerSystem</w:Selector>
            </w:SelectorSet>
        </a:ReferenceParameters>
    </p:ServiceProvided>
    <p:UserOfService>
        <a:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</a:Address>
        <a:ReferenceParameters>
            <w:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
            </w:ResourceURI>
            <w:SelectorSet>
                <w:Selector Name="Name">ManagedSystem</w:Selector>
                <w:Selector Name="CreationClassName">CIM_ComputerSystem</w:Selector>
```

```
            </w:SelectorSet>
          </a:ReferenceParameters>
      </p:UserOfService>
</p:CIM_AssociatedPowerManagementService>
</wsman:Results>
```

A kapott eredményből számunkra fontos a `p:UserOfService` elemen belüli részfa, az itteni `a:ReferenceParameters` elemmel tudunk később hivatkozni a felügyelt rendszert reprezentáló *CIM_ComputerSystem*-re.

3. Hívjuk meg a *CIM_PowerManagementService* objektum *RequestPowerStateChange* metódusát! A metódus 4 paramétert vár, ebből kettő kötelező. A *PowerState* bemenő paraméternél egy egész számot kell átadni a PowerState felsorolt típus szerint. Az API referenciából kiderül, hogy az enumeráció 2-es értéke jelenti, a „Power On” műveletet. A *ManagedElement* bemenő paraméternél pedig az `a:ReferenceParameters` részfát kell átadni. Emiatt viszont az összes paramétert egy XML fájlban kell megadnunk, amit a kérés előtt össze kell állítanunk:

```
<p:RequestPowerStateChange_INPUT
xmlns:p="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem"
xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing">
   <p:PowerState>2</p:PowerState>
   <p:ManagedElement>
    <a:ReferenceParameters>
     <w:ResourceURI>
         http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
     </w:ResourceURI>
     <w:SelectorSet>
      <w:Selector Name="Name">ManagedSystem</w:Selector>
      <w:Selector Name="CreationClassName">CIM_ComputerSystem</w:Selector>
     </w:SelectorSet>
    </a:ReferenceParameters>
   </p:ManagedElement>
 </p:RequestPowerStateChange_INPUT>
```

A kérés elküldése:

```
winrm invoke RequestPowerStateChange http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_PowerManagementService -remote:http://localhost/wstrans/pro/eoi20/10.40.120.130/wsman
-a:Basic -u:admin -p:AMTje1szo% -encoding:utf-8 -format:pretty -file:poweron.xml
```

Ennek eredményeképpen a felügyelt gép elindul.

A fenti példához hasonló módon célszerű a dokumentációban leírt lépéseket követve végezni más, bonyolultabb műveleteket. Meg kell jegyezni, hogy az XML fájlokkal való kézi műveletek a `winrm` hiányosságai miatt kellenek. A PowerShell 2.0 ennél lényegesen kényelmesebb mechanizmust ad WS-Man használatára, ám sajnos az autentikációnál a felhasználói név elé helyez egy ’\’ karaktert (feltételezve, hogy a másik végpont is Windows, amelyik megérti, hogy ez az Active Directory Domaint jelöli), emiatt viszont e sorok írásakor a PowerShell 2.0 az AMT-vel még egyáltalán nem volt használható.

1.7 Ellenőrző kérdések

1. Mit jelent a távfelügyelet az informatikai rendszerek estében, és mi indokolja az elterjedését?
2. Soroljon fel a távfelügyelet konkrét feladatai közül legalább hármat.
3. Soroljon fel a távfelügyelet szoftveres és hardveres eszközei közül párat. Mik a szoftveres megoldások jellemzői/előnyei/hátrányai? Mik a hardveres megoldásoké?
4. Mik a wake-on-LAN megoldás hátrányai?
5. Jellemezze az Intel vPro platform két elemét.
6. Milyen üzemmódokban használható az Intel AMT? Mi a különbség közöttük?
7. Miket köt össze az Intel AMT virtuális soros csatornája, és mire használható?

8. Milyen esetben használható a hálózati forgalom átirányítása, és mire jó?
9. Mi az ágensek jelenlét-ellenőrzésének feladata, és hogyan működik?
10. Mi az Intel AMT DTK? Egy-egy mondatban ismertesse legalább három komponensét.
11. Mi a különbség a SOAP Web Services és a WS-Man interfészek között?
12. Mi a WS-Translator szerepe? Winrm használatánál hogyan adjuk meg a Translatort futtató gép elérhetőségét, és hogyan a kérés valódi címzettjét?