Centar Informacijske Sigurnosti

DEP zaštita

srpanj 2011.

CIS-DOC-2011-07-018

Upozorenje

Podaci, informacije, tvrdnje i stavovi navedeni u ovom dokumentu nastali su dobrom namjerom i dobrom voljom te profesionalnim radom CIS-ovih stručnjaka, a temelje se na njihovom znanju i petnaestak godina iskustva u radu u informacijskoj sigurnosti. Namjera je da budu točni, precizni, aktualni, potpuni i nepristrani.

Ipak, oni su dani samo kao izvor informacija i CIS ne snosi nikakvu izravnu ili posrednu odgovornost za bilo kakve posljedice nastale korištenjem podataka iz ovog dokumenta.

Ukoliko primijetite bilo kakve netočnosti, krive podatke ili pogreške u ovom dokumentu, ili imate potrebu komentirati sadržaj molimo Vas da to javite elektroničkom poštom na adresu info@CIS.hr.

O CIS-u

CIS izrađuje pregledne dokumente (eng. white paper) na teme iz područja informacijske sigurnosti koji će biti korisni zainteresiranoj javnosti, a u svrhu **podizanje njezine svijesti o informacijskoj sigurnosti** i **sposobnosti za čuvanje i zaštitu informacija i informacijskih sustava**. Pored toga, CIS razvija i održava mrežni portal www.CIS.hr kao referalnu točku za informacijsku sigurnost za cjelokupnu javnost; izrađuje obrazovne materijale namijenjene javnosti; organizira događaje za podizanje svijesti o informacijskoj sigurnosti u javnosti i pojedinim skupinama te djeluje u suradnji sa svim medijima.

CIS **okuplja mlade** zainteresirane za informacijsku sigurnost i radi na njihovom pravilnom odgoju i obrazovanju u području informacijske sigurnosti te pripremu za **profesionalno bavljenje informacijskom sigurnošću**.

Centar informacijske sigurnosti [CIS] nastao je 2010. godine na poticaj Laboratorija za sustave i signale[LSS] Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu, a kao posljedica 15togodišnjeg rada na istraživanju, razvoju i primjeni informacijske sigurnosti. LSS je među ostalim potaknuo osnivanje CARNetovog CERTa i sudjelovao u izradi Nacionalnog programa informacijske sigurnosti RH.

Smisao CISa je da bude **referentno mjesto za informacijsku sigurnost** za javnost, informatičare i posebno za mlade te da sustavno podiže njihovu svijest i sposobnosti u području informacijske sigurnosti.

Rad CISa podržava Ministarstvo znanosti, obrazovanja i sporta Republike Hrvatske, a omogućuju sponzori.

Prava korištenja

Sadržaj

1. UVOD 4

2. DEP 5

2.1. SKLOPOVSKA IZVEDBA 5

2.2. PROGRAMSKA IZVEDBA 6

2.3. POSTAVKE DEP ZAŠTITE NA RAČUNALU 6

2.3.1. Provjera pomoću kontrolne ploče 6

2.3.2. Provjera pomoću komandne linije 7

2.3.3. Provjera pomoću grafičkog sučelja 8

2.3.4. Aktiviranje šireg spektra programske DEP zaštite 9

3. NX I XD BIT 12

3.1. TABLICA STRANICA 12

3.2. SUSTAVI ZAŠTITE DRUGIH OPERACIJSKIH SUSTAVA 12

4. PROBLEMI DEP ZAŠTITE 13

4.1. SLUČAJNI ODABIR POSTAVE ADRESNOG PROSTORA 13

4.2. STRUKTURIRANO UPRAVLJANJE IZNIMKAMA 13

4.3. OBAVEZNA KONTROLA CJELOVITOSTI 15

4.4. POZNATA OGRANIČENJA 16

5. ZAKLJUČAK 17

6. LEKSIKON POJMOVA 18

7. REFERENCE 20

1. Uvod

Zaštita memorije je način kontrole prava pristupa memoriji na računalu i sastavni je dio većine današnjih operacijskih sustava. Glavna svrha zaštite je onemogućavanje procesima da pristupaju memoriji koja im nije dodijeljena. Na taj način procesi ne mogu djelovati jedni na druge kao ni na sam operacijski sustav.

Jedan od čestih načina zaštite memorije je segmentacija, koja pretpostavlja podjelu memorije računala u manje dijelove. Na primjer, arhitektura x86 (32-bitna računala) koristi dvije tablice za adresiranje segmenata u memoriji računala: globalnu opisnu tablicu (eng. Global Descriptor Table) i lokalnu opisnu tablicu (eng. Local Descriptor Table). Pritom globalna tablica sadrži zapise o segmentima koji su dostupni svim procesima dok lokalna sadrži zapise o segmentima koji su privatni i dostupni tek dodijeljenim procesima. Već samim odvajanjem privatnih od javnih, podaci u memoriji su donekle zaštićeni.

Segmentacija nije savršena zaštita, ona je samo navedena kao primjer jednostavnog mehanizma zaštite. Najbolja opcija za računalo bi bila istovremena sklopovska i programska zaštita pri čemu bi se njihove funkcije međusobno komplementirale. Upravo to je način na koji je osmišljena DEP zaštita.

Ovaj dokument opisuje način na koji DEP štiti računalo te mehanizme koji nadopunjuju njegove slabosti. Na jednostavan način će se opisati način na koji korisnik može sam promijeniti postavke na svom računalu kao i što će te promjene uzrokovati. Spomenuti će se i poznati problemi s DEP zaštitom na koje su korisnici kroz godine naišli te će se dati upute kako riješiti ili zaobići te probleme.

2. DEP

DEP (eng. *Data Execution Prevention*) ili zaštita od izvršavanja podataka je način zaštite prisutan kod većine operacijskih sustava današnjice. Svrha mu je spriječiti aplikacije i usluge da pokreću programski kod u dijelu memorije gdje to nije dopušteno tako da se taj dio označi kao „neizvršiv“ (eng. *non-executable memory*). Na taj način se sprečavaju problemi poput prilično čestog zlonamjernog iskorištavanja preljeva spremnika (eng. *buffer overflow*)[1].

Slika 1 ilustrira problem preljeva spremnika. U ovom slučaju promatramo 2 varijable – A i B, od kojih varijabla A ima veličinu 8 okteta, a B 2 okteta. Ako bi neki program htio u varijablu A upisati riječ *'excessive*' bez provjeravanja duljine te riječi, višak bi se „prelio“ u susjedni spremnik. Na taj način bi se promijenio podatak u varijabli B. Za slučaj da je u varijabli B pohranjen neki kritični podatak operacijskog sustava, sustav ne bi mogao ispravno funkcionirati. Upravo zato je potrebna zaštita memorije.

Ime varijable	A								B	
Vrijednost	[null string]								1979	
Vrijednost u hex formatu	00	00	00	00	00	00	00	00	07	BB

Ime varijable	A								B	
Vrijednost	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	25856	
Vrijednost u hex formatu	65	78	63	65	73	73	69	76	65	00

Slika 1. *Preljev spremnika (eng. buffer overflow)*

DEP je, pod tim nazivom, sastavni dio Windowsa od inačice XP SP2. Drugi operacijski sustavi koriste drugo ime za isti princip zaštite (kao što je opisano u poglavlju 3.2).

DEP zaštita ima i sklopovski i programski aspekt. Sklopovlje (odnosno procesor) označava u kojem dijelu memorije nije dopušteno izvršavanje programskog kôda. Prilikom nedozvoljenog pokušaja izvođenja, procesor dojavljuje pogrešku. Nakon toga programski dio DEP-a treba ugasiti aplikaciju ili uslugu zbog koje je došlo do dojave pogreške.

Računala koja nemaju potrebno sklopovlje (uglavnom starija od 2005. godine) još uvijek mogu podržavati programsku izvedbu DEP-a, iako je učinkovitost zaštite u tom slučaju manja.

2.1. Sklopovska izvedba

Sklopovski dio DEP zaštite zahtjeva posebne opcije u procesoru. Procesor označava memoriju atributom koji pokazuje da se iz tog dijela memorije ne smije pokretati programski kôd. Cijela memorija se smatra neizvršnom osim ako je posebno napomenuto drugačije. Posljednjih godina AMD-ovi i Intelovi čipovi dolaze s integriranom podrškom za DEP, no računala starija od 2005. godine je uglavnom nemaju.

DEP označava pojedine stranice virtualne memorije[2] (eng. *virtual memory page*) u zapisu tablice stranica (eng. *Page Table Entry*) posebnim bitom čime se razlikuje izvršna od neizvršne memorije:

- AMD procesori koriste NX (eng. *No Execute*) bit za označavanje stranica sa zabranom izvršavanja kôda,

[1] Preljev spremnika se događa kad programi, dok zapisuju podatke u dodijeljeni spremnik, prijeđu granice spremnika te upisuju podatke u susjednu memoriju. Takva situacija može biti problematična, pogotovo u situaciji kad je u susjednoj memoriji zapisan neki kritični podatak potreban za ispravan rad operacijskog sustava.

[2] Kad računalo nema dovoljno radne memorije (RAM) da podrži sve pokrenute procese, odvaja se dio memorije tvrdog diska te se on koristi kao pomoćna (virtualna) radna memorija. Dijelovi te memorije se nazivaju stranicama, a u Windows operacijskom sustavu se pohranjuju pod nazivom *pagefile.sys*.

- Intelovi procesori koriste XD (eng. *Execute Disable*) bit kojim se onemogućava izvršavanje kôda.

Arhitektura procesora određuje kako će DEP biti primijenjen u sklopovlju i kako će se označavati stranice virtualne memorije. No bez obzira na izvedbu, procesor će dojaviti pogrešku tek kad neka aplikacija ili usluga pokuša izvršiti kôd u nedozvoljenom dijelu memorije.

Da bi koristio ove značajke, procesor mora biti u PAE[3] načinu rada (eng. *Physical Address Extension*), kojeg će Windows automatski omogućiti kako bi podržao DEP zaštitu.

Tri su metode kojima korisnik može provjeriti da li ima osposobljen DEP na računalu:

- provjeravanje preko kontrolne ploče,
- provjeravanje pomoću komadne linije te
- putem grafičkog sučelja.

2.2. Programska izvedba

Programska izvedba DEP zaštite je oblikovana tako da blokira zloćudni kôd koji iskorištava mehanizme upravljanja iznimkama u operacijskom sustavu Windows. Microsoft ovaj oblik DEP zaštite naziva „sigurnosnim strukturiranim upravljanjem iznimkama“ (eng. *Safe Structured Exception Handling*) ili *SafeSEH*. Kad se pojavi iznimka u izvršavanju nekog programa, *SafeSEH* provjerava da li je ta iznimka registrirana u funkcijskoj tablici dotične aplikacije te da li aplikacija traži izvršavanje unatoč dojavljenoj iznimci.

Programski izvedeni DEP radi na bilo kojem računalu koje može pokrenuti operacijski sustav Windows XP. Standardno, on štiti samo određene datoteke sustava, neovisno o sklopovskim DEP mogućnostima procesora. Korisnik može odabrati širi spektar zaštite, ali ova opcija nije postavljena kao standardna zbog starijih ili loše napisanih programa koji koriste podatke iz neizvršne memorije prilikom izvršavanja (što DEP blokira). Noviji programi su nešto „inteligentniji“ te izbjegavaju miješanje izvršnih i neizvršnih dijelova memorije.

2.3. Postavke DEP zaštite na računalu

Sljedeći korak u shvaćanju DEP zaštite je provjeravanje kako je DEP primjenjen na vlastitom računalu. Kao za većinu toga u operacijskim sustavima Windows, i za provjeru DEP postavki postoji nekoliko različitih načina, a na čitatelju je da izabere njemu najprikladniji. U svim narednim primjerima je korišteno računalo bazirano na Intel 64-bitnoj arhitekturi s operacijskim sustavom Windows 7 te je, po potrebi, dan administratorski pristanak za pojedine akcije.

2.3.1. Provjera pomoću kontrolne ploče

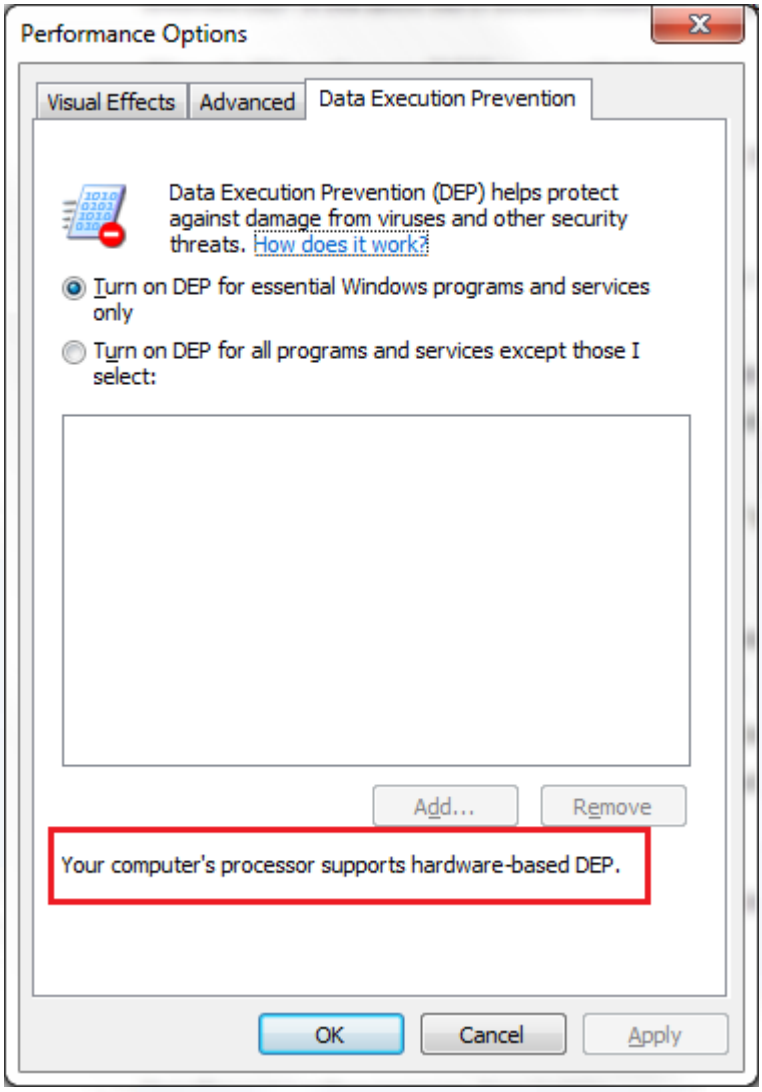
Pomoću kontrolne ploče (eng. *Control Panel*) je najjednostavnije provjeriti DEP postavke i ovaj način zahtjeva najmanje tehničkog znanja. Potrebno je slijediti korake, dajući pritom administratorski pristanak gdje je potrebno (Windows Vista i 7):

1. Windows XP: Kliknuti *Start* → desni klik mišem na *Moje Računalo* (eng. *My Computer*);
   Windows Vista i 7: Kliknuti *Start* → desni klik mišem na *Računalo* (eng. *Computer*),
2. odabrati *Svojstva* (eng. *Properties*),
3. Windows XP: Odabrati polje *Dodatno* (eng. *Advanced*);
   Windows Vista i 7: Odabrati *Dodatne Postavke* (eng. *Advanced System Settings),*
4. u odjeljku *Performansa* (eng. *Performance*) kliknuti na *Postavke* (eng. *Settings*) te

[3] PAE je dodatna mogućnost IA32 procesora da se adresira više od 4GB radne memorije (što inače predstavlja ograničenje za 32-bitna računala).

5. odabrati polje *Data Execution Prevention* čime se otvara prozor poput onog na slici Slika 2. Računalo u primjeru na istoj slici ima ugrađenu sklopovsku podršku za DEP što je vidljivo iz poruke u dnu prozora.

Ovom metodom se mogu vidjeti općenite postavke – da li je DEP zaštita podržana te da li je uključena. Za više informacija je potrebno koristiti jednu od idućih metoda.



Slika 2. *Prozor s opcijama DEP zaštite*

2.3.2. Provjera pomoću komandne linije

Korištenjem alata *Wmic* u komandnoj liniji se mogu provjeriti DEP postavke. Potrebno je slijediti korake:

1. Windows XP: Kliknuti *Start* → *Run* → upisati „cmd“ i pritisnuti tipku *Enter*.
   Windows Vista i 7: Kliknuti *Start* → upisati „cmd“ u polje za pretragu i pritisnuti tipku *Enter*.
2. U komandnu liniji upisati:

```
C:\WIndows\system32> wmic OS Get.
```

Kao rezultat naredbe će se ispisati podaci o računalu, a polja koja se tiču DEP zaštite na isprobanom računalu su prikazana u tablici (Tablica 1). Iz tablice se može očitati da je DEP zaštita omogućena te aktivirana (jer su vrijednost svih polja TRUE).

Polje	Vrijednost polja
DataExecutionPrevention_32BitApplications	TRUE
DataExecutionPrevention_Available	TRUE
DataExecutionPrevention_Drivers	TRUE
DataExecutionPrevention_SupportPolicy	2

Tablica 1. *Postavke DEP zaštite*

Moguće vrijednosti i značenja polja **DataExecutionPrevention_SupportPolicy** su dane u tablici (Tablica 2). Na isprobanom računalu je vrijednost polja bila '2' ili *'OptIn'*

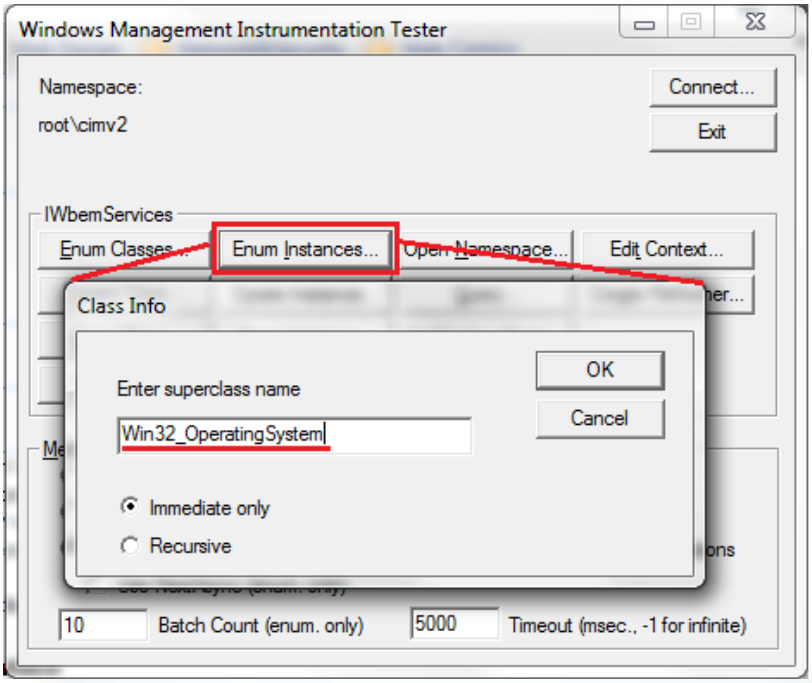
Vrijednost polja DataExecution Prevention_Su pportPolicy	Zaštita	Opis
2	OptIn	Samo određene usluge i dijelovi sustava Windows su pod DEP zaštitom. Standardna postavka za Windows XP, Vista i 7. Vrijedi iznimka za 64-bitne arhitekture gdje su sve aplikacije standardno zaštićene osim ako se izričito ne zatraži gašenje zaštite za aplikaciju.
3	OptOut	DEP je omogućen za sve procese. Administratori mogu ručno napraviti popis pojedinih aplikacija koje ne žele pod DEP zaštitom. Ova je opcija standardna za WIndows Server 2003 SP1.
1	AlwaysOn	DEP je omogućen za sve procese. Zanemaruju se eventualni zahtjevi za gašenjem DEP zaštite pojedinih aplikacija i procesa.
0	AlwaysOff	DEP nije omogućen ni za koji proces. Zanemaruju se eventualni zahtjevi za DEP zaštitom.

Tablica 2. *Vrijednosti i značenja polja DataExecutionPrevention_SupportPolicy*

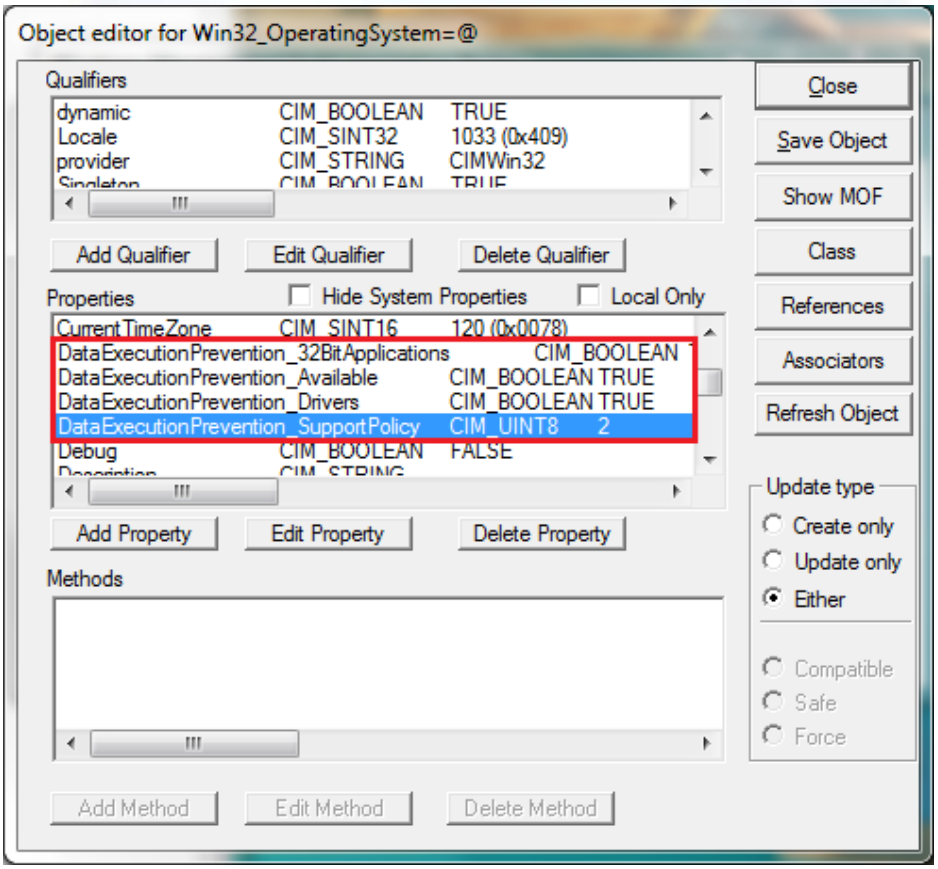
2.3.3. Provjera pomoću grafičkog sučelja

Da bi se pomoću grafičkog sučelja provjerile postavke, potrebno je slijediti korake:

1. Windows XP: Kliknuti *Start* → *Run* → upisati „wbemtest“ i pritisnuti tipku *Enter*.
   Windows Vista i 7: Kliknuti *Start* → upisati „wbemtest“ u polje za pretragu i pritisnuti tipku *Enter*.
2. U *Windows Management Instrumentation Tester* prozoru kliknuti gumb *Connect...*
3. U polje *Namespace* upisati „root/cimv2“ i kliknuti gumb *Connect*.
4. Kliknuti gumb *Enum Instances...*
5. Upisati „*Win32_OperatingSystem*“ u polje za tekst i kliknuti gumb *OK* (Slika 3).
6. U dobivenom prozoru dvostrukim klikom odabrati opciju pri vrhu (koja počinje s „*Win32_OperatingSystem*“).
7. U prozoru *Object Editor* koji se otvorio pronaći svojstvo *DataExecutionPrevention_Available* u području *Svojstva* (eng. *Properties*) (Slika 4).
8. Dvostrukim klikom odabrati *DataExecutionPrevention_Available*.
9. Obratiti pažnju na vrijednost polja (TRUE/FALSE).



Slika 3. *Koraci 4 i 5 u korištenju grafičkog sučelja*

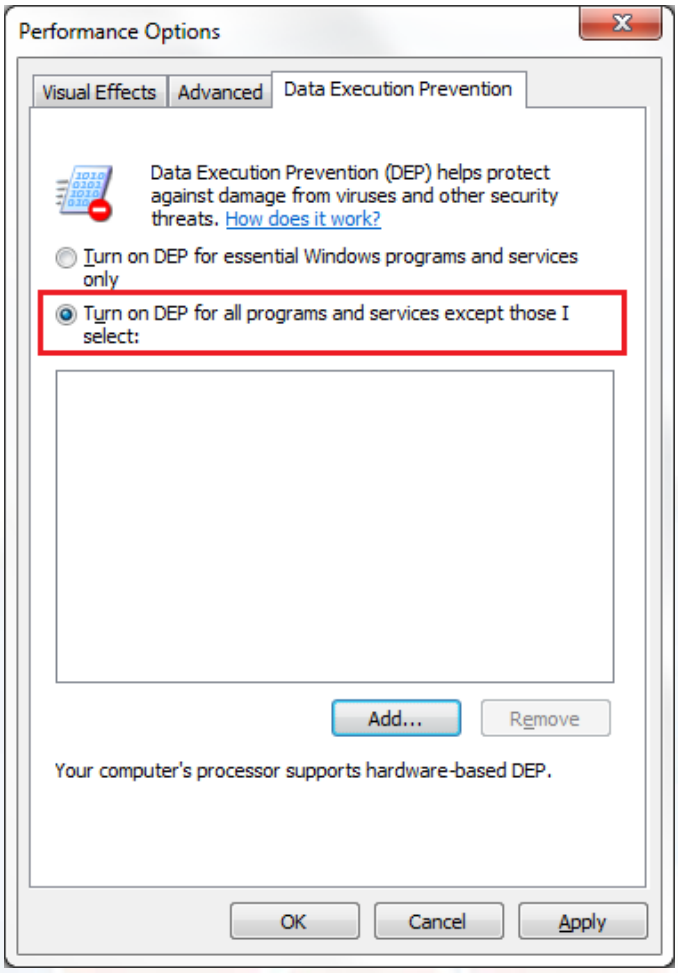


Slika 4. *Korak 7 u korištenju grafičkog sučelja*

2.3.4. Aktiviranje šireg spektra programske DEP zaštite

Kako bi se aktivirao širi spektar programske izvedbe DEP zaštite, potrebno je za početak otvoriti prozor s DEP opcijama slijedeći korake u potpoglavlju 2.3.1. U prozoru (Slika 5) se

treba odabrati druga ponuđena opcija prilikom čega se u okviru ispod mogu odabrati alati za koje korisnik ne želi da budu pod DEP zaštitom.



Slika 5. *Mijenjanje DEP postavki*

Treba napomenuti da 64-bitni sustav Windows automatski nameće DEP zaštitu svim 64-bitnim alatima, no ne i 32-bitnim alatima. Osim već spomenutih načina provjere DEP postavki na računalu, naredbom:

```
C:\Windows\system32> bcdedit
```

u komandnoj liniji (s omogućenim administratorskim ovlastima) moguće je provjeriti podatke o računalu te stanje NX bita – čije vrijednosti su one iz Tablice 2 (*OptIn, OptOut, AlwaysOn, AlwaysOff*).

Dobra praksa je aktivirati način rada *OptOut* u kojem su sve aplikacije pod DEP zaštitom, a tek se za pojedine može napomenuti da ne moraju biti pod zaštitom. To se može izvesti naredbom:

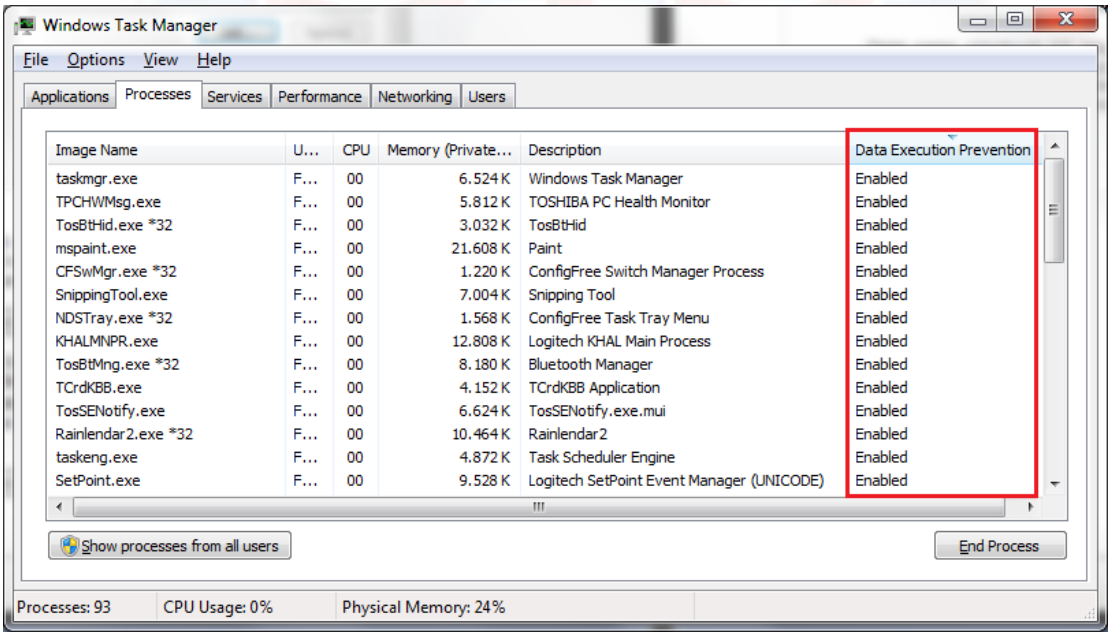
```
C:\Windows\system32> bcdedit.exe /set nx OptOut.
```

Na slici (Slika 6) se može vidjeti dio ispisa naredbe 'bcdedit' prije i nakon mijenjanja načina rada DEP-a. Crvenom bojom je označena razlika u vrijednostima NX bita.

```
osdevice                partition=C:
systemroot              \Windows
resumeobject
nx                      OptIn

C:\Windows\system32>bcdedit.exe /set nx OptOut
The operation completed successfully.

C:\Windows\system32>bcdedit

Windows Boot Manager
--------------------
identifier              {bootmgr}
device                  partition=\Device\HarddiskVolume1
description             Windows Boot Manager
locale                  en-US
inherit                 {globalsettings}
default                 {current}
resumeobject
displayorder            {current}
toolsdisplayorder       {memdiag}
timeout                 30

Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \Windows\system32\winload.exe
description             Windows 7
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence
recoveryenabled         Yes
osdevice                partition=C:
systemroot              \Windows
resumeobject
nx                      OptOut
```

Slika 6. *Ispis naredbe 'bcdedit'*

Osim same vrijednosti NX bita, nakon promjene načina rada u *OptOut* sve 32-bitne aplikacije su također došle pod DEP zaštitu. Slika 7 prikazuje prozor upravitelja zadacima (eng. *Task Manager*) u kojem je jasno vidljivo da je za svaku aplikaciju omogućena DEP zaštita (eng. *enabled*).



Slika 7. *Upravitelj zadacima*

3. NX i XD bit

NX bit (eng. *No eXecute*) je sastavni dio AMD procesorske tehnologije koji pomaže u odvajanju memorije za pohranu podataka i za pohranu procesorskih instrukcija. AMD ovu mogućnost zove 'naprednom zaštitom od virusa' (eng. *Enhanced Virus Protection*).

Operacijski sustav s podrškom za NX bit može označiti određene dijelove memorije kao neizvršne nakon čega će procesor odbiti izvršavati programske kôdove koji se eventualno nalaze u tom dijelu memorije. Ova tehnika se koristi za zaštitu računala od zlonamjernih alata koji bi ubacili te izvršili svoj kôd u dijelu memorije u kojem drugi proces pohranjuje podatke (iskorištavanjem preljeva spremnika).

Istu funkcionalnost kao i NX bit imaju i druge arhitekture – Intel ovaj bit naziva XD (eng. *eXecute Disable*), a ARM XN (eng. *eXecute Never*), iako je pojam 'NX bit' postao generični opis za bitove svih arhitektura s istom funkcijom.

3.1. Tablica stranica

Tablica stranica (eng. *page table*) je podatkovna struktura koju koriste virtualni memorijski sustavi na računalima kako bi pohranili veze između virtualnih i fizičkih adresa. Virtualne adrese su jedinstvene za pojedini proces, a fizičke adrese su jedinstvene u odnosu na cjelokupnu radnu memoriju.

NX bit je zapravo bit najveće važnosti (bit 63) u 64-bitnim zapisima u tablici stranica. Ako je postavljen u 0, na toj stranici se može pokretati kôd. Ako je postavljen u 1, ta se stranica koristi samo za pohranu podataka.

Kao što je već spomenuto, NX bit se koristi samo uz PAE format tablice stranica (koji dopušta adresiranje sa 64 bita umjesto sa samo 32). U protivnom tablica ima samo 32-bitne podatke pa ne postoji bit 63 koji bi utjecao na zaštitu memorije.

3.2. Sustavi zaštite drugih operacijskih sustava

Iako se o DEP-u mnogo raspravlja u računalnoj zajednici, operacijski sustav Windows nije jedini koji implementira taj tip zaštite. Drugi operacijski sustavi također koriste mogućnosti NX bita.

- **FreeBSD**

  Operacijski sustav FreeBSD je počeo podržavati NX bit na x86-64 i x86 procesorima u lipnju 2004. godine.

- **Linux**

  Jezgra Linux podržava NX bit na procesorima x86-64 i x86 koji su ga sposobni koristiti. To obuhvaća trenutne 64-bitne procesore AMD, Intel, Transmeta i VIA.

  Podrška za NX bit je na procesorima x86-64 dodana 2004. godine te je nešto kasnije iste godine dodana i podrška za 32-bitni način rada 64-bitnih procesora.

- **Mac OS X**

  Sustav Mac OS X podržava NX bit na svim Intelovim procesorima (od inačice 10.4.4 nadalje).

- **Android**

  Od inačice 2.3, Android koristi arhitekturu koja standardno koristi neizvršne stranice.

4. Problemi DEP zaštite

Iako otežava izvođenje pojedinih tipova napada, DEP sam po sebi nije dostatan za potpuno osiguranje od svih tipova napada. Zbog toga se treba kombinirati s drugim sigurnosnim mogućnostima kao što su slučajni odabir postave adresnog prostora (eng. *address space layout randomization* – ASLR), zaštita od prepisivanja strukturiranog upravljanja iznimkama (eng. *structured exception handler overwrite protection* – SEHOP) i obavezna kontrola cjelovitosti (eng. *mandatory integrity control*). Ove sigurnosne tehnike i DEP se međusobno nadopunjuju – jedan pokriva slabosti drugog te ih upravo zato treba kombinirano koristiti.

4.1. Slučajni odabir postave adresnog prostora

Slučajni odabir postave adresnog prostora ili ASLR (eng. *Address Space Layout Randomization*) je tehnika kojom se ključna podatkovna područja razmještaju nasumice po adresnom prostoru procesa. Na taj način se sprečavaju napadi koji se oslanjaju na pronalazak određenih podataka u memoriji prije izvršavanja. Na primjer, napadač koji je pohranio izvršni kôd na određeni stog sa željom da se taj kôd pokrene prilikom paljenja računala, suočiti će se s problemom pronalaska stoga. Naravno, uvijek može pogađati adresu na kojoj se nalazi, ali pogrešan izbor adrese često dovodi do „rušenja“ računala nakon čega su adrese ponovno izmiješane. Što je veći adresni prostor, to je manja vjerojatnost da će napadač „od oka“ pogoditi adresu, što znači da je pojačana sigurnost.

Operacijski sustavi koji podržavaju ASLR su:

- **OpenBSD** – prvi operacijski sustav s podrškom za ASLR,
- **Linux** – nešto slabija inačica ASLR-a,
- **MS Windows** – podržava ASLR od siječnja 2007., a Windows 7 nudi mogućnost gašenja ASLR i DEP zaštite,
- **Mac OS X** – nudi neku vrstu nasumičnog adresiranja, ali bez svih mogućnosti ASLR-a, pretpostavlja se da će MAC OS X 10.7 Lion (datum izlaska na tržište 20.7.2011.) imati potpunu podršku za ASLR,
- **iOS** – uveden ASLR od inačice iOS 4.3.

4.2. Strukturirano upravljanje iznimkama

Ono što se u kontekstu ove teme naziva iznimkom (ili pogreškom) je događaj koji se može pojaviti tijekom izvođenja programa prilikom čega se zahtjeva izvršavanje kôda izvan normalnog kontrolnog toka. Postoje 2 tipa iznimki:

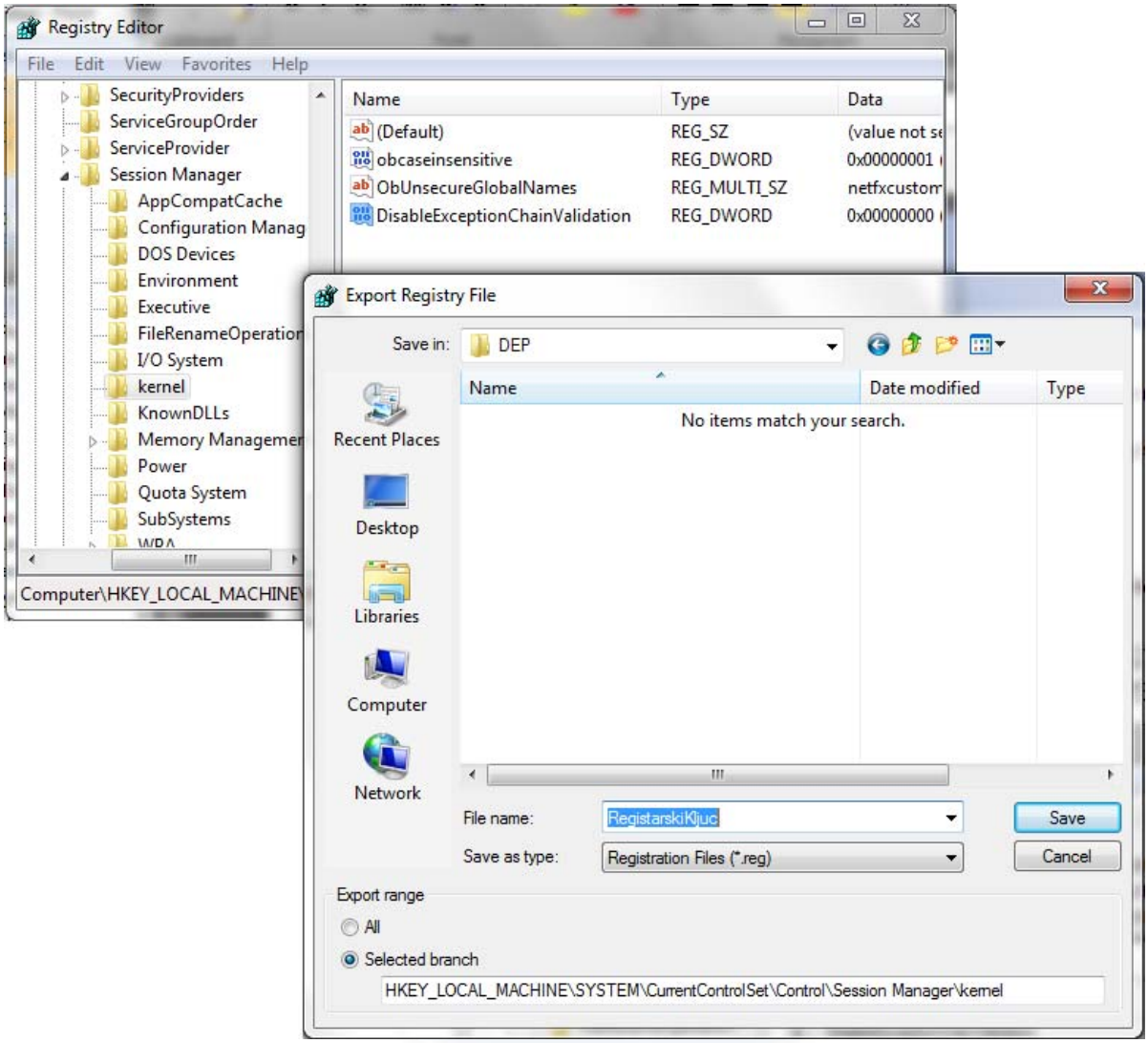
- sklopovske iznimke - pokreće procesor prilikom izvođenja određenih nizova instrukcija kao što su na primjer dijeljenje s nulom ili pokušaj pristupa neispravnim adresama u memoriji.
- programske iznimke - pokreću isključivo aplikacije ili operacijski sustav. Na primjer, sustav može reagirati prilikom unošenja neispravnih varijabli.

SEH (eng. *Structured Exception Handler*) je mehanizam za upravljanje iznimkama oba tipa. SEH pruža korisniku potpunu kontrolu nad upravljanjem iznimkama, te pruža podršku programima za pronalaženje pogrešaka (eng. *debugger*), a može se koristiti sa svim programskim jezicima i na svim računalima.

SEHOP (eng. *SEH Overwrite Protection*) je mehanizam koji štiti SEH mehanizam od toga da ga neki drugi podatak prepiše ili izmjeni. SEHOP je standardno omogućen u operacijskom sustavu Windows Server 2008, a onemogućen u sustavima Vista i 7. Aktiviranje SEHOP-a je u nekoliko slučajeva znalo blokirati izvođenje određenih aplikacija kao što su Skype, Cygwin i aplikacije zaštićene Armadillom.

Prilikom ručne aktivacije SEHOP-a potrebno je vrlo pažljivo slijediti korake. Preporuča se napraviti i kopiju registra za svaki slučaj. Kopija registra se može napraviti na sljedeći način:

1. Kliknuti *Start* → upisati „*regedit*“ u polje za pretragu i pritisnuti *Enter*. Dati administratorsko odobrenje ako je potrebno.
2. Pronaći registarski ključ kojeg se želi sačuvati i jednom kliknuti mišem na njega.
3. U izborniku *File* pdabrati opciju *Export*.
4. U izborniku *Save in* odabrati lokaciju gdje se želi pohraniti kopija registarskog ključa te upisati ime za njega u polju *File name*.
5. Kliknuti *Save*.



Slika 8. *Pohranjivanje sigurnosne kopije registra prije provođenja ikakvih promjena*

Konačno, da bi se aktivirao SEHOP, potrebno je slijediti korake:

1. Kliknuti *Start* → upisati „*regedit*“ u polje za pretragu i pritisnuti tipku *Enter*
2. Pronaći podključ registra:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\kernel\DisableExceptionChainValidation
```

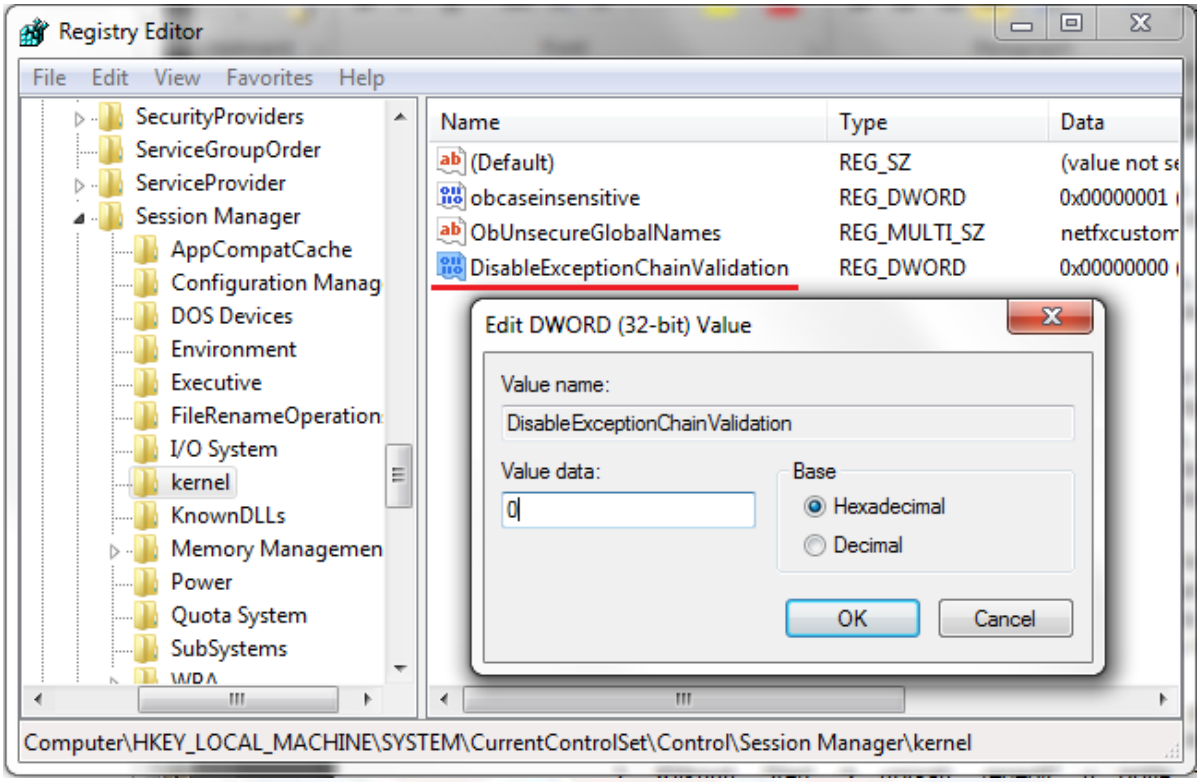
Ako registarski ključ ne postoji, potrebno je pronaći podključ:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\kernel\
```

te:

a. kliknuti desnom tipkom miša na mapi *kernel*, odabrati *New* te kliknuti *DWORD Value,*
b. upisati „*DisableExceptionChainValidation“* te pritisnuti tipku *Enter*.

3. Dvostruko kliknuti mišem na *DisableExceptionChainValidation*.
4. Promijeniti vrijednost registra (polje *Value data*) u '0' (Slika 9). Na taj način se aktivira SEHOP mehanizam. Vrijednost '1' bi ga deaktivirala.
5. Izaći iz uređivača registra.



Slika 9. *Mijenjanje vrijednosti registarskog ključa*

4.3. Obavezna kontrola cjelovitosti

Obavezna kontrola cjelovitosti ili MIC (eng. *Mandatory Integrity Controls*) je jezgrena sigurnosna mogućnost prisutna u operacijskom sustavu Windows od inačica Vista i Server 2008. Ona može selektivno ograničiti pristupne dozvole pojedinim programima ili programskim komponentama u kontekstima gdje ih se smatra manje vjerodostojnima u odnosu na ostale vjerodostojnije kontekste koji se odvijaju na istom korisničkom računu. Windows Vista definira 4 razine integriteta:

- niska – procesi moraju biti eksplicitno napravljeni za ovu razinu,
- srednja – ovu razinu imaju procesi koje je pokrenuo „običan“ korisnik (bez administratorskih ovlasti),
- visoka – razina procesa koje je pokrenuo administrator,
- sustav – procesi koje je pokrenuo sustav.

MIC dodjeljuje razine privilegija koje onemogućavaju procesima nižih razina da pokreću procese više razine. Dobar primjer razina integriteta na operacijskom sustavu Windows su web preglednici Internet Explorer inačica 7 i 8. Oni se mogu pokrenuti u „Zaštićenom načinu“ na Visti i kasnijim inačicama sustava Windows. U ovoj konfiguraciji proces iexplore.exe se odvija s niskom razinom integriteta kako bi se smanjio njegov utjecaj na sustav. Na taj način on ne može mijenjati objekte na najvišoj razini sustava.

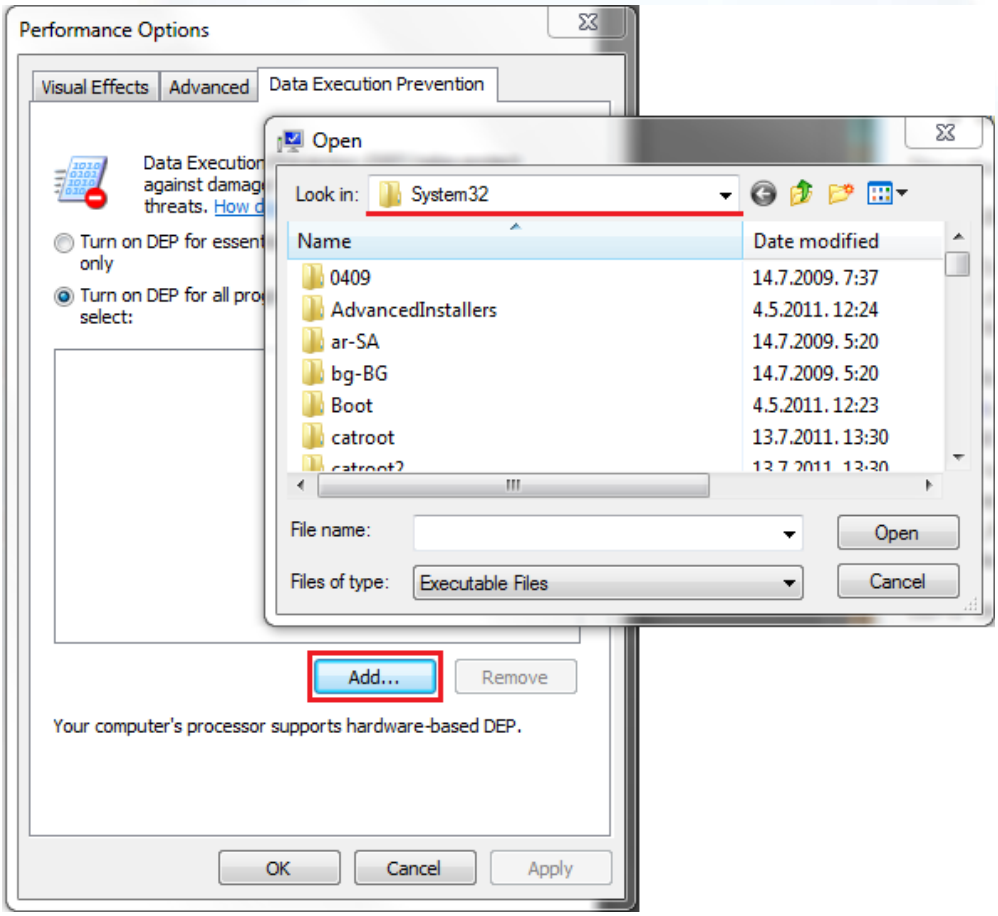
4.4. Poznata ograničenja

DEP je uvijek omogućen za 64-bitne aplikacije i to se ne može isključiti. Naspram toga, neke provjereno dobroćudne 32-bitne aplikacije rade na način kojeg će DEP blokirati. Poznati primjer toga je bio Internet Explorer 7 na operacijskom sustavu Windows Vista. U tom slučaju je potrebno blokirane aplikacije staviti na *OptOut* popis (Slika 10):

1. Slijedeći korake u poglavlju 2.3.1 doći do prozora za upravljanje DEP opcijama.
2. Kliknuti mišem na gumb „*Add*...“
3. U ponuđenom izborniku odabrati aplikaciju kojoj se želi dopustiti rad bez DEP zaštite.

Računalo korišteno u ovom primjeru je 64-bitne arhitekture, zbog čega ne postoji samo mapa „*System*“ u koju se pohranjuju sve aplikacije već postoje 2 mape: „*SysWOW64*“ u kojoj se nalaze 64-bitne i „*System32*“ u kojoj se nalaze 32-bitne aplikacije. Može se primjetiti kako je izbornik na slici automatski smjestio korisnika u mapu „*System32*“ s 32-bitnim aplikacijama. Između ostalog, to je zato što se 64-bitne aplikacije ne mogu staviti na *OptOut* popis.

Prilikom stavljanja problematične aplikacije na *OptOut* popis, poželjno je provjeriti da aplikacija nije štetna. Prvo bi trebalo provjeriti na stranici proizvođača aplikacije da li postoji novija inačica koja je kompatibilna s DEP načinom zaštite. Na nekim tehničkim forumima se čak preporučuje da se nikako ne isključuje DEP zaštita osim ako ne postoji način da se preživi bez aplikacije u pitanju.



Slika 10. *Odabir aplikacija za koje se želi da rade mimo DEP zaštite*

5. Zaključak

DEP zaštita se osniva na sinergiji između sklopovlja i programskog dijela računala. Kada sklopovlje dojavi iznimku, operacijski sustav će odlučiti kako će se nositi s njom. Iako računala starija od 2005. godine uglavnom nemaju sklopovlje koje podržava DEP zaštitu, programski dio zaštite još uvijek postoji.

DEP zaštita, odnosno NX bit, predstavlja veliki korak naprijed po pitanju računalne sigurnosti. Iako sama po sebi ne pruža potpunu zaštitu, u kombinaciji s drugim metodama zaštite pruža prilično pouzdan sigurnosni mehanizam u borbi protiv napada na računalni sustav.

U ovom dokumentu je pokazano kako se mogu aktivirati dodatne zaštite koje dolaze s operacijskim sustavom Windows inačice XP SP2 i novijima. To uključuje dodatne opcije DEP programske zaštite te SEHOP.

Kao što je slučaj i s poznavanjem prijetnji na Internetu, privatni korisnici računala bi trebali znati što je više moguće o načinu na koji njihovo računalo funkcionira. Na taj način se mogu spriječiti gubici podataka, a i uštediti na uslugama računalnih servisa.

6. Leksikon pojmova

Virus - Računalni virus

Virusi su programi koji se mogu kopirati i napasti računalo bez znanja ili dopuštenja korisnika, na razne načine (preko Internet-a, CD-a, USB-a...). Virusi dolaze većinom s drugim programima, kao što su npr. trojanski konji, kako bi maskirali svoj rad i kako bi ih bilo još teže za otkriti. Namjene virusa su različite, mogu služiti samo kako bi radili štetu no neki su manje štetni i samo usporavaju računalo i smetaju korisniku u radu. Virusi se spremaju u memoriju računala i pokreću se s operacijskim sustavom i inficiraju programe koji se pokreću.

http://www.ust.hk/itsc/antivirus/general/whatis.html

http://os2.zemris.fer.hr/ns/2008_Mackovic/virusi.htm

IA-32 - Intelova 32-bitna procesoska arhitektura

Intelova 32-bitna procesorska arhitektura predstavlja skup naredbi za najrašireniji mikroprocesor organizacije Intel. To je 32-bitno proširenje x86 procesorske arhitekture, a prvi mikroprocesor koji je se zasnivao na ovoj arhitekturi je Intel 80386.

http://www.pctechguide.com/ia-32-intel-architecture-32-base-instruction-set-for-32-bit-processors
http://pc.wikia.com/wiki/Intel_Architecture_32-Bit

NX bit

NX bit (od eng. *No eXecute*) je tehnologija koju koriste procesori proizvođača AMD kako bi se označilo da je dio memorije rezerviran za procesorske instrukcije (programski kôd) ili za pohranu podataka. Drugi proizvođači procesora koriste druga imena za bitove s jednakom funkcijom kao i NX bit. Intel tako koristi pojam **XD** (eng. *Execute Disable*) , a ARM **XN** (eng. eXecute Never). Ponekad se „NX“ koristi kao univerzalni pojam za sve bitove te funkcije.

http://en.wikipedia.org/wiki/NX_bit

http://hardware.earthweb.com/chips/article.php/3358421

SEH – Structured Exception Handling

SEH ili strukturirano upravljanje iznimkama je mehanizam na kojeg se sustav oslanja prilikom dizanja sklopovske ili programske iznimke. Drugim riječima, kad se u sustavu dogodi nešto neočekivano (npr. preljev, neovlašteni pristup), SEH određuje kako će se računalo nositi s tom iznimkom (npr. gašenjem aplikacije, gašenjem računala).

http://msdn.microsoft.com/en-us/library/ms680657(v=vs.85).aspx

SEHOP – Structured Exception Handling Overwrite Protection

SEHOP je mehanizam za zaštitu od aplikacija koje pokušavaju zapisati podatke preko SEH mehanizma. Ova opcija se uključuje prilikom paljenja računala tako da štiti sve aplikacije neovisno o aktiviranim opcijama.

http://ssj100.fullsubject.com/t137-structured-exception-handling-overwrite-protection-sehop

SafeSEH – Safe Structured Exception Handling

Opcija pokretanja SEH mehanizma u sigurnom načinu rada. Kad se pojavi iznimka u izvršavanju neke aplikacije, SafeSEH provjerava da li je ta iznimka registrirana u funkcijskoj tablici dotične aplikacije te da li aplikacija traži izvršavanje unatoč dojavljenoj iznimci.

http://msdn.microsoft.com/en-us/library/9a89h429.aspx

ASLR – Address Space Layout Randomization

Slučajni odabir postave adresnog prostora ili ASLR (eng. *Address Space Layout Randomization*) je sigurnosna računalna tehnika koja pohranjuje ključni kôd bitan za ispravan rad sustava na različite lokacije u memoriji prilikom svakog paljenja računala. Na taj se način otežavaju napadi koji se pokreću zajedno s računalom te pokušavaju dohvatiti podatke bitne za rad sustava jer su oni svaki put nasumice razmješteni.

http://blogs.msdn.com/b/michael_howard/archive/2006/05/26/address-space-layout-randomization-in-windows-vista.aspx

http://en.wikipedia.org/wiki/Address_space_layout_randomization

MIC – Mandatory Integrity Controls

Obavezna kontrola cjelovitosti ili MIC (eng. *Mandatory Integrity Controls*) je mehanizam koji kontrolira pristup pojedinim osiguranim objektima. Koriste se oznake razina integriteta kako bi se osiguralo da aplikacije s nižom razinom integriteta ne mogu pristupiti objektima (podacima) više razine.

http://en.wikipedia.org/wiki/Mandatory_Integrity_Control

http://msdn.microsoft.com/en-us/library/bb648648(v=vs.85).aspx

7. Reference

[1] Wikipedia: Memory Protection, http://en.wikipedia.org/wiki/Memory_protection, srpanj 2011.

[2] Wikipedia: Executable Memory Protection, http://en.wikipedia.org/wiki/Executable_space_protection, srpanj 2011.

[3] Wikipedia: Buffer Overflow, http://en.wikipedia.org/wiki/Buffer_overflow, srpanj 2011.

[4] Microsoft Help and Support: RAM, virtual memory, pagefile, and memory management in Windows, http://support.microsoft.com/kb/2160852/en-us, prosinac 2010.

[5] Computer Education: Data Execution Prevention (DEP), http://vlaurie.com/computers2/Articles/dep.htm, srpanj 2011.

[6] Microsoft Help and Support: A detailed description of the Data Execution Prevention (DEP) feature in Windows XP Service Pack 2, Windows XP Tablet PC Edition 2005, and Windows Server 2003, http://support.microsoft.com/kb/875352/en-us, rujan 2006.

[7] Microsoft Help and Support: How to determine that hardware DEP is available and configured on your computer, http://support.microsoft.com/kb/912923, listopad 2007.

[8] Windows: Data Execution Prevention: frequently asked questions, http://windowshelp.microsoft.com/Windows/en-US/help/186de3d0-01af-4d4c-981d-674637d2f4bf1033.mspx, srpanj 2011.

[9] Hardware Central: CPU-Based Security: The NX Bit, http://www.hardwarecentral.com/reviews/article.php/3358421/CPU-Based-Security-The-NX-Bit.htm, svibanj 2004.

[10] Wikipedia: NX bit, http://en.wikipedia.org/wiki/NX_bit, srpanj 2011.

[11] Wikipedia: Address space layout randomization, http://en.wikipedia.org/wiki/Address_space_layout_randomization, srpanj 2011.

[12] Microsoft Help and Support: How to enable Structured Exception Handling Overwrite Protection (SEHOP) in Windows operating systems, http://support.microsoft.com/kb/956607, lipanj 2011.

[13] Microsoft Help and Support: Back up the registry, http://windows.microsoft.com/en-us/windows7/Back-up-the-registry, srpanj 2011.

[14] Wikipedia: Mandatory Integrity Control, http://en.wikipedia.org/wiki/Mandatory_Integrity_Control, srpanj 2011.