กรณีศึกษาการป้องกันการโจมตีผ่านเครือข่ายโดยอาศัยช่องว่างของบัฟเฟอร์ ด้วยเทคโนโลยีการหยุดยั้งการประมวลผลบนซีพียู

ศิริชัย รุจิพัฒนพงศ์ และ สุทธิพันธุ์ แสนละเอียด***

บทคัดย่อ

NX (No Executed) Bit เปิดตัวพร้อมกับซีพียู AMD รุ่นใหม่ เป็นเทคโนโลยีที่ทำงานร่วมกับคุณสมบัติ Data Execution Prevention (DEP) ของวินโดวส์ XP Service Pack 2 (SP2) และวินโดวส์ Server 2003 SP1 ประโยชน์หลักของ DEP คือช่วยป้องกันมิให้มีการเรียกใช้คำสั่งจากเพจ ข้อมูล ตามปกติคำสั่งจะไม่ทำงานจาก Heap และ Stack โดย NX Bit และ DEP จะตรวจหาคำสั่งที่ทำงานจากตำแหน่งเหล่านี้ และทำให้เกิดข้อผิดพลาดเมื่อมีการเรียกใช้คำสั่ง จากผลการทดลองเมื่อใช้ซีพียูที่รองรับ NX Bit ร่วมกับคุณสมบัติ DEP ของวินโดวส์ XP SP2 พบว่าสามารถหยุดการทำงานของคำสั่งที่เข้าถึง Stack ได้แต่ NX Bit ยังมีปัญหาในการทำงานร่วมกับโปรแกรม ในปัจจุบัน เช่น Snagit เป็นต้น ส่งผลให้ผู้ใช้นิยมปิดการทำงาน NX Bit จึงไม่เกิดประโยชน์ในแง่ของการใช้งานจริง

คำสำคัญ : NX XD DEP AMD64 สแตก

1. บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กลวิธีในการโจมตีผู้ใช้ในระบบเครือข่ายที่ได้รับความนิยมมักจะใช้วิธีการทำให้บัฟเฟอร์ล้น (Buffer Overflow) ด้วยการใช้คำสั่งหรือส่งหนอนไวรัสหรือโทรจันเข้าสู่คอมพิวเตอร์ของผู้ใช้ และยิงคำสั่ง (Process) ให้กับซีพียูของคอมพิวเตอร์เพื่อประมวลผลแบบตลอดเวลา อันจะทำให้เกิดปัญหาบัฟเฟอร์ล้นและซีพียูหยุดการทำงานไปเสี้ยววินาที ช่วงเวลานี้เองนักโจมตีหรือแครกเกอร์จะส่งชุดคำสั่งเพื่อเข้าควบคุมการทำงาน โดยผู้ใช้คอมพิวเตอร์ไม่ทราบหรือสังเกตการทำงานได้เลย

ด้วยปัญหาเหล่านี้ผู้ผลิตซีพียูอย่างอินเทล (Intel) และเอเอ็มดี (AMD) จึงได้ออกแบบให้ซีพียูสามารถตัด (Kill) การทำงานของโปรแกรมหรือชุดคำสั่งที่เป็นอันตรายต่อเครื่องคอมพิวเตอร์ ซีพียูจะเพิ่มบิตสำหรับตรวจสอบที่เรียกว่า NX (No Execute) Bit ติดไปกับโปรเซสที่ทำงาน บิต NX จะคอยตรวจจับการเรียกใช้ตำแหน่งของหน่วยความจำที่สงวนไว้วงจร NX Bit จะสั่งตัดการทำงานของคำสั่งทันที เพื่อมิให้เกิดปัญหาการรันโปรแกรมแฝงอันจะนำไปสู่การโจมตีได้ ซีพียูของอินเทลจะเรียกเทคโนโลยีนี้ว่า XD (Execute Disable) Bit ซึ่งมีกลไกการทำงานเหมือนกัน

สำหรับไมโครซอฟท์ได้สนับสนุนแนวทางการป้องกันปัญหาด้วยเช่นกัน โดยออกชุดปรับปรุง Service Pack 2 (SP2) เพื่อสนับสนุนการทำงานร่วมกับ NX Bit โดยการทำงานจะแบ่งออกเป็น 2 รูปแบบตามซีพียูที่ใช้งาน หากซีพียูสนับสนุน NX Bit ตัววินโดวส์จะปล่อยให้ซีพียูเป็นผู้ควบคุม แต่ถ้าเป็นกรณีของซีพียูรุ่นเก่าวินโดวส์จะช่วยเสริมการทำงานตรงจุดนี้ให้ด้วยซอฟท์แวร์ ดังนั้นแนวทางในการทำวิจัยจะเป็นการเปรียบเทียบผลลัพธ์ระหว่างเครื่องคอมพิวเตอร์ที่ไม่มีระบบป้องกันใดๆ, เครื่องคอมพิวเตอร์ที่ใช้คุณสมบัติการป้องกันด้วย Windows XP SP2 และเครื่องคอมพิวเตอร์ที่ใช้ซีพียูรุ่นใหม่ที่มีฟังก์ชัน NX Bit เพื่อวิเคราะห์ผลลัพธ์ในการป้องกันระบบ

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษากลไกการทำงานของเทคโนโลยี NX Bit ของซีพียู

1.2.2 เพื่อเปรียบเทียบความสามารถในการป้องกันปัญหาบัฟเฟอร์ด้วย NX Bit

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ปัญหาที่ส่งผลให้เกิดการโจมตีทางเครือข่าย

ปัญหาที่ส่งผลให้เกิดการโจมตีทางเครือข่ายสามารถเกิดได้จากตัวผู้ใช้งานและตัวระบบ ปัญหาทางฝั่งผู้ใช้งานมักเกิดจากความบกพร่องหรือมักง่าย จึงตั้งรหัสผ่านที่ง่ายต่อการเดาหรือไม่มีการเข้ารหัสคำสั่งที่เป็นความลับ ในขณะที่ปัญหาตัวระบบอาจเกิดจากช่องโหว่ของตัวซอฟท์แวร์ระบบ การส่งโปรแกรมสอดแนม หรือโปรแกรมที่ออกแบบมาเพื่อโจมตีเครือข่าย โดยส่งผลให้เกิดช่องโหว่ของระบบให้ผู้ไม่ประสงค์ดีเข้าถึงระบบได้ง่าย

* *ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ*

** *คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ*

2.2 สถาปัตยกรรม AMD64

บริษัท AMD ได้ออกแบบซีพียู 64 บิตของตนขึ้นมาเพื่อแข่งขันด้านการตลาดกับบริษัท Intel โดยมีชื่อเรียกว่า AMD64 สถาปัตยกรรมของ AMD64 จะแตกต่างกับ IA64 ของ Intel โดยสิ้นเชิง แม้ว่ากลไกของ AMD64 จะประมวลผลแบบ RISC เหมือนกัน แต่ AMD64 ได้ออกแบบให้มีวงจรสำหรับแปลง (Translation) ชุดคำสั่งจาก CISC ไปเป็น RISC จึงสามารถใช้งานร่วมกับโปรแกรมทั่วไปได้อย่างไม่มีปัญหา ซีพียู AMD64 จะทำงานในโหมด Legacy เมื่อใช้งานร่วมกับระบบปฏิบัติการ 32 บิตเดิม และสามารถเปลี่ยนไปทำงานในโหมด Long เพื่อรองรับ 64 บิตบนระบบปฏิบัติการ 64 บิตได้

2.2.1 โหมดการทำงานของ AMD64

สถาปัตยกรรม x86 เดิมรองรับโหมดการทำงานเพียง 4 แบบคือ Real Mode, Protected Mode, Virtual-8086 Mode และ System Management Mode [1] โดยสถาปัตยกรรม AMD64 ก็รองรับโหมดการทำงานแบบเดิมเช่นกัน ทั้งยังรองรับโหมดการทำงานแบบใหม่ที่เรียกว่า Long Mode ดังตารางที่ 1 แสดงความแตกต่างของโหมด Long และ Legacy

Long Mode ประกอบด้วยโหมดย่อย 2 โหมด คือ 64-bit Mode และ Compatibility Mode สำหรับโหมด 64 บิตรองรับฟังก์ชันใหม่หลายอย่าง รวมถึงการอ้างอิงพื้นที่ตำแหน่งเสมือนแบบ 64 บิต ในขณะที่โหมด Compatibility จะรองรับการทำงานร่วมกับโปรแกรม 16 และ 32 บิตเดิม ๆ บนระบบปฏิบัติการ 64 บิต การทำ งานในโหมด Long ระบบปฏิบัติการจะต้องเปิดทำงานในโหมด Protected ก่อน

Legacy Mode ประกอบด้วยโหมดย่อย 3 โหมด คือ Real Mode, Protected Mode และ Virtual-8086 Mode โหมด Legacy นอกจากจะรองรับการทำงานร่วมกับแอพพลิเคชัน 16 และ 32 บิตแล้ว ยังสามารถทำงานบนระบบปฏิบัติการ 16 และ 32 บิตเดิมได้ด้วย

2.2.2 Page Translation กับการทำงานของ NX Bit

สถาปัตยกรรม x86 เดิมรองรับการแปลงตำแหน่งเสมือน 32 บิตไปยังตำแหน่งทางกายภาพ 32 บิต (ตำแหน่งทางกายภาพ 36 หรือ 40 บิตรองรับในโหมดพิเศษ) สถาปัตยกรรม AMD64 ได้ขยายการรองรับตำแหน่งเสมือน 64 บิตไปเป็นตำแหน่งทางกายภาพขนาด 52 บิต ตำแหน่งเสมือนจะถูกแปลงไปเป็นตำแหน่งทางกายภาพโดยอาศัยตารางการแปลงแบบลำดับชั้น (Hierarchical) ซึ่งถูกสร้างและจัดการด้วยระบบปฏิบัติการ

ตารางที่ 1 แสดงโหมดการทำงานของสถาปัตยกรรม AMD64

Mode		System Software Required	Application Recompile Required	Defaults[1] Address Size (bits)	Defaults[1] Operand Size (bits)	Register Extensions[2]	Maximum GPR Width (bits)
Long Mode[3]	64-Bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16	16		
Legacy Mode	Protected Mode	Legacy 32-bit OS	no	32	32	no	32
				16	16		
	Virtual-8086 Mode			16	16		32
	Real Mode	Legacy 16-bit OS					

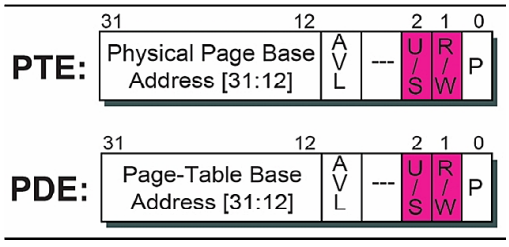
Note:
1. Defaults can be overridden in most modes using an instruction prefix or system control bit.
2. Register extensions includes eight new GPRs and eight new XMM registers (also called SSE registers).
3. Long mode supports only x86 protected mode. It does not support x86 real mode or virtual-8086 mode.

ตารางที่ 2 แสดงการแปลงเพจและตำแหน่งในโหมดของ AMD64

Mode		Physical-Address Extensions (CR4.PAE)	Page-Size Extensions (CR4.PSE)	Page-Directory Page Size	Resulting Physical-Page Size	Maximum Virtual Address	Maximum Physical Address
Long Mode	64-Bit Mode	Must be enabled	–	PDE.PS=0	4 Kbyte	64-bit	52-bit
	Compatibility Mode			PDE.PS=1	2 Mbyte		
Legacy Mode		Enabled	–	PDE.PS=0	4 Kbyte	32-bit	52-bit
				PDE.PS=1	2 Mbyte		52-bit
		Disabled	Disabled	–	4 Kbyte		32-bit
			Enabled	PDE.PS=0	4 Kbyte		32-bit
				PDE.PS=1	4 Mbyte		40-bit

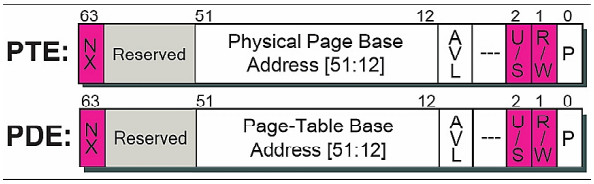
2.2.2.1 Page Translation ของซีพียู 32 บิต

ด้วยข้อจำกัดของกลไก Page-Protection ใน PTE (Page Table Entry) และ PDE (Page Directory Entry) ซึ่งมีตัวแปร U/S และ R/W เป็นตัวควบคุมกลไกการอ่าน/เขียนอยู่แล้ว จึงไม่สามารถควบคุมการอ่านและเขียนเพจแบบอิสระได้ตามต้องการ [2]



ภาพที่ 1 โครงสร้างของ x86 Page-Protection

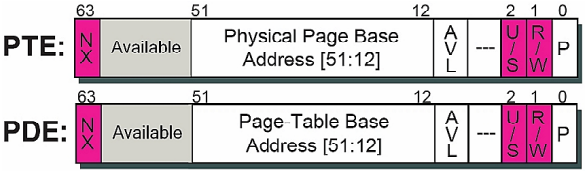
การออกแบบเพจใหม่ในลักษณะของ PAE (Physical Address Extension) โดยขยายเพจให้มีขนาด 64 บิต แต่รองรับการอ้างอิงตำแหน่งจริงเพียง 52 บิตแรกและบิต NX จะอยู่ที่ตำแหน่งบิตที่ 64



ภาพที่ 2 โครงสร้างของ x86 Page-Protection แบบ PAE

2.2.2.2 Page Translation ของซีพียู AMD64

การใช้งานซีพียู 64 บิตบน Windows XP x64 Edition จะมีการอ้างอิงเพจแบบ 64 บิต ดังนั้นเพจ PTE และ PDE บนซีพียู 64 บิตจะมีขนาดใหญ่พอที่จะเพิ่ม NX Bit สำหรับตรวจสอบเข้าไปได้ สำหรับวินโดวส์ 64 บิตรุ่นธรรมดาจะอ้างอิงตำแหน่งจริง 52 บิตแรก (0-51) เท่านั้น โดยสงวนตำแหน่งช่วง 52-62 ไว้รองรับในอนาคต NX Bit ที่เพิ่มเข้าไปจะทำงานเป็นอิสระเพื่อตรวจสอบการทำงานในส่วนของ No-Execute Memory หากมีการเรียกใช้หน่วยความจำส่วนที่ Protected จะแจ้งเตือนและตัดการทำงานทันที และผู้ใช้ก็ไม่สามารถยกเลิกการทำงานของ NX Bit ได้



ภาพที่ 3 โครงสร้างของ AMD64 Page-Protection

2.3 การจัดการหน่วยความจำของวินโดวส์ XP

ตามปกติโปรเซสบนวินโดวส์ 32 บิตมีขนาดเสมือนเท่ากับ 2 GB ถ้าอิมเมจถูกกำหนดให้ทำงานแบบ Large Address Space ตัวระบบจะบู๊ตด้วยคำสั่งพิเศษซึ่งจะขยายขนาดเสมือนเป็น 3 GB บนวินโดวส์ 32 บิต และ 4 GB บนวินโดวส์ 64 บิต ขนาดตำแหน่งโปรเซสเสมือนบนวินโดวส์ 64 บิตบนระบบ x64 จะมีขนาดสูงสุด 8,192 GB (ในอนาคต) ขนาดของหน่วยความจำกายภาพที่วินโดวส์รองรับมีตั้งแต่ 2 GB ถึง 1,024 GB ขึ้นอยู่กับรุ่นของวินโดวส์

2.3.1 No Execute Page Protection

วินโดวส์ XP Service Pack 2 และวินโดวส์ Server 2003 Service Pack 1 สามารถทำงานร่วมกับโปรเซสเซอร์ที่รองรับคุณสมบัติทางฮาร์ดแวร์ที่เรียกว่า “การป้องกันแบบ No Execute” ซึ่งจะอยู่ในโปรเซสเซอร์ AMD64 (AMD Athlon 64 และ AMD Opteron) และโปรเซสเซอร์ 32 บิตของ AMD (AMD Sempron บางรุ่น) รวมไปถึง Intel IA-64 (Itanium) และโปรเซสเซอร์ Intel Pentium 4 and Xeon ที่มีคุณสมบัติ Intel Extended Memory 64 Technology (EM64T)

การป้องกันเพจแบบ No Execute (หรืออีกชื่อหนึ่งคือ Data Execution Prevention : DEP) เป็นการจัดการคำสั่งในเพจที่ถูกกำหนดให้เป็น No Execute ซึ่งจะช่วยป้องกันไวรัสจากบัคที่มักจะเกิดขึ้นในตัวระบบ เมื่อมีการเข้าถึงเพจที่ No Execute ในโหมดเคอร์เนล ตัวระบบจะหยุดการทำงานด้วยคำสั่งในการตรวจสอบบัก ATTEMPTED_EXECUTE_OF_NOEXECUTE_MEMORY หรือถ้าอยู่ในโหมดผู้ใช้ก็จะแสดงข้อผิดพลาด STATUS_ACCESS_VIOLATION (0xc0000005) [2]

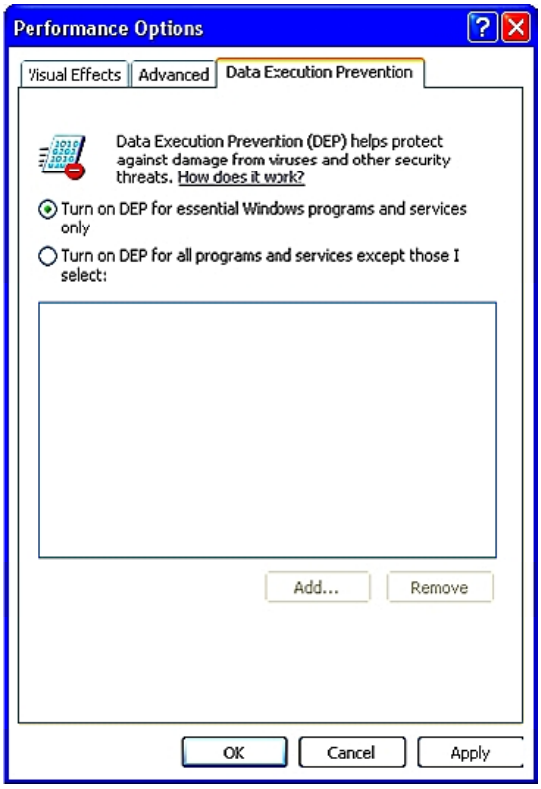
สำหรับวินโดวส์ 64 บิตการป้องกัน No Execute มีผลกับโปรแกรมและไดรเวอร์ 64 ทั้งหมดและผู้ใช้ไม่สามารถยกเลิกได้ บนวินโดวส์ 64 บิต No Execute จะมีผลกับส่วน Thread Stacks (ทั้งโหมดเคอร์เนลและโหมดผู้ใช้) Kernel Paged Pool และ Kernel Session Pool ในขณะที่วินโดวส์ 32 บิต No Execute จะมีผลกับ Thread Stacks และเพจโหมดผู้ใช้

การป้องกัน No Execute ในส่วนของแอพพลิเคชัน 32 บิตขึ้นอยู่กับคำสั่ง /NOEXECUTE ในไฟล์ Boot.ini การตั้งค่านี้สามารถปรับเปลี่ยนได้จากแท็ป Data Execution Prevention ภายใต้ My Computer > Properties > Advanced > Performance Settings เมื่อมีการตั้งค่าNo Execute ในส่วนของหน้าต่าง DEP จะมีการเข้าไปแก้ไขคำสั่ง /NOEXECUTE ในไฟล์ Boot.ini โดยอัตโนมัติดังตารางที่ 3 ในกรณีที่ต้องการยกเลิกการทำงานของ No Execute บนแอพพลิเคชัน 32 บางโปรแกรม สามารถเข้าไปแก้ไขรีจิสตรีโดยตรงจากคีย์ HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\AppCompatFlags\Layers โดยเพิ่มตำแหน่ง(Path) ของโปรแกรมในคำสั่ง DisableNXShowUI

ค่าปกติของการตั้งค่าเพื่อใช้งานคุณสมบัติ No Execute ของวินโดวส์ XP SP2 คือ /NOEXECUTE=OPTIN คือให้มีผลเฉพาะตัวแอพพลิเคชันและเซอร์วิสของวินโดวส์เท่านั้น โดยจะไม่ส่งผลกับโปรแกรมอื่นที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ แต่กรณีของวินโดวส์ Server 2003 SP1 ค่าปกติจะเป็น /NOEXECUTE=OPTOUT คือมีผลกับทุกโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ [2]

ตารางที่ 3 แสดงวิธีการจัดการส่วนควบคุม No Execute

คำสั่งใน Boot.ini	ตัวเลือก DEF	การทำงาน
/NOEXECUTE =OPTIN	Turn on DEP for necessary Windows programs and services only	เปิดการทำงาน NX Bit กับโปรแกรมและบริการ (Service) ของ Windows XP SP2
/NOEXECUTE =OPTOUT	Turn on DEP for all programs and service except those that I select	เปิดการทำงาน NX Bit กับโปรแกรมทั้งหมด แต่ผู้ใช้เลือกปิด NX Bit บางโปรแกรมได้
/NOEXECUTE =ALWAYSON	-	เปิดการทำงาน NX Bit กับโปรแกรมทั้งหมด โดยไม่สามารถควบคุมหรือยกเลิกได้
/NOEXECUTE =ALWAYSOFF	-	ปิดการทำงาน NX Bit ทั้งหมด



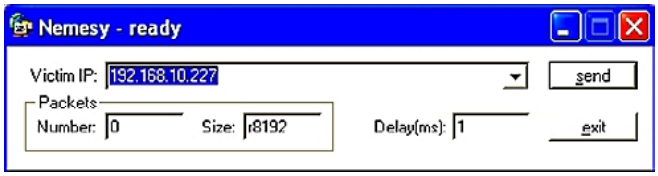
ภาพที่ 4 แสดงหน้าต่างสำหรับการตั้งค่า Data Execution Prevention (DEP)

ค่าปกติของการตั้งค่าเพื่อใช้งานคุณสมบัติ No Execute ของวินโดวส์ XP SP2 คือ /NOEXECUTE=OPTIN คือ ให้มีผลเฉพาะตัวแอพพลิเคชันและเซอร์วิสของวินโดวส์เท่านั้น โดยจะไม่ส่งผลกับโปรแกรมอื่นที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ แต่กรณีของวินโดวส์ Server 2003 SP1 ค่าปกติจะเป็น /NOEXECUTE=OPTOUT คือมีผลกับทุกโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ [2]

3. วิธีการดำเนินงาน

ตามทฤษฎีการทำงานของ NX Bit เพื่อจุดประสงค์ในการป้องกันปัญหาบัฟเฟอร์ล้น (Buffer Overflow) จึงได้ออกแบบการทดสอบออกเป็น 3 รูปแบบคือ

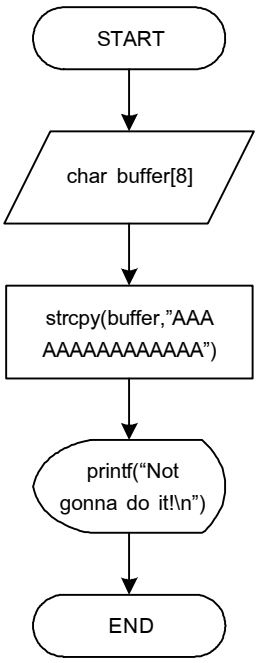
3.1 การทดสอบด้วยโปรแกรม Nemesy ตัวโปรแกรมออกแบบมาเพื่อทดสอบการส่งแพ็กเก็ตผ่านเครือข่ายไปยังเครื่องปลายทาง เสมือนเป็นการโจมตีผ่านทางเครือข่ายคอมพิวเตอร์



ภาพที่ 5 แสดงโปรแกรม Nemesy

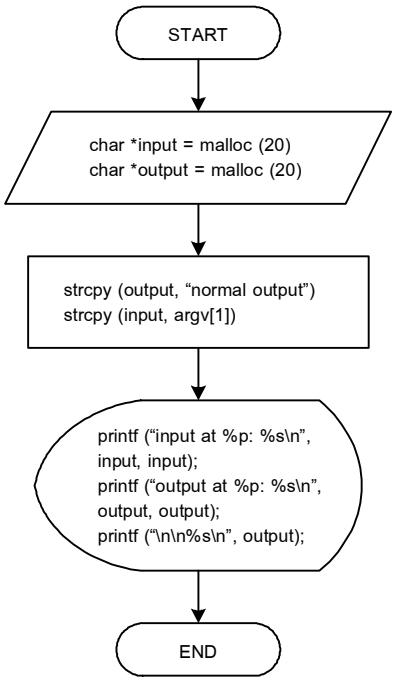
3.2 การทดสอบด้วยโปรแกรมที่มีปัญหา Overflow จะเป็นการเขียนโปรแกรมด้วยภาษา C โดยเป็นโปรแกรมที่จงใจทำให้เกิดปัญหา Overflow เพื่อดูว่า Windows XP สามารถจัดการกับโปรแกรมเหล่านี้อย่างไร

3.2.1 โปรแกรมทดสอบ Stack Overflow



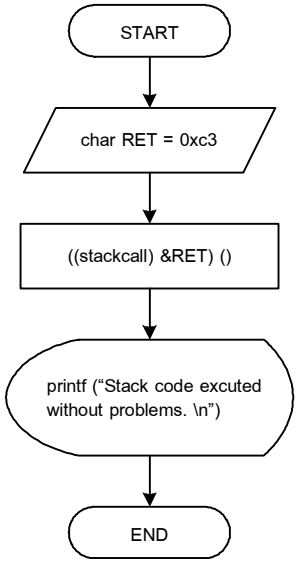
ภาพที่ 6 แสดงโปรแกรมทดสอบ Stack Overflow

3.2.2 โปรแกรมทดสอบ Heap Buffer Overflow



ภาพที่ 7 แสดงโปรแกรมทดสอบ Heap Overflow

3.3 โปรแกรมทดสอบ NX Bit ตามลักษณะการทำงานของ NX Bit จะป้องกันการอ้างอิงตำแหน่งของหน่วยความจำที่มีการ Protected ไว้ ดังนั้นทดสอบด้วยการเขียนโปรแกรมโดยใช้คำสั่ง Stackcall เข้าไปอ่านยังตำแหน่งของหน่วยความจำโดยตรง เพื่อดูผลการทำงานของ NX Bit



ภาพที่ 8 แสดงโปรแกรมทดสอบ NX Bit

4. ผลการทดลอง

4.1 สภาพแวดล้อมที่ใช้ในการทดสอบระบบ

4.1.1 เครื่องคอมพิวเตอร์จำนวน 2 เครื่อง เครื่องแรกใช้ซีพียู Athlon 64 ซึ่งสนับสนุน NX Bit ส่วนเครื่องที่สองเป็นซีพียู Athlon XP ที่ไม่สนับสนุน NX Bit คอมพิวเตอร์ทั้งสองเครื่องจะติดตั้งเฉพาะไดรเวอร์ที่จำเป็นเท่านั้น และไม่มีการติดตั้งโปรแกรมป้องกันไวรัสหรือสปายแวร์ใดๆ

4.1.2 ระบบปฏิบัติการ Windows XP โดยทดสอบกับรุ่น SP1 และ SP2 ที่สนับสนุน NX Bit เพื่อดูผลการทำงานร่วมกับระบบปฏิบัติการ

4.2 ผลการทดสอบ

ผลการทดสอบพบว่าการทดสอบด้วยโปรแกรม Nemesy พบว่า Windows XP SP1 ไม่สามารถจัดการปัญหาเหล่านี้ได้ แม้ว่าโปรแกรม Nemesy จะเป็นโปรแกรมที่รู้จักกันมานานแล้ว และมีผู้ไม่ประสงค์ดีเคยนำไปใช้งานจริง แต่ Windows XP SP1 ก็ไม่สามารถป้องกันได้

ตารางที่ 5 สรุปผลการทดลองกรณีโปรแกรมที่ส่งแพ็กเก็ตผ่านเครือข่าย

ระบบปฏิบัติการ	ประเภทซีพียู	ผลการทดลอง
Windows XP SP1	ไม่มี NX Bit	ทำงานได้ตามปกติ
	รองรับ NX Bit	ทำงานได้ตามปกติ
Windows XP SP2	ไม่มี NX Bit	ไม่สามารถทำงานได้
	รองรับ NX Bit	ไม่สามารถทำงานได้

ตารางที่ 6 สรุปผลการทดลองกรณีโปรแกรมที่มีปัญหา Stack Overflow และ Heap Buffer Overflow

ระบบปฏิบัติการ	ประเภทซีพียู	ผลการทดลอง
Windows XP SP1	ไม่มี NX Bit	ไม่สามารถทำงานได้
	รองรับ NX Bit	ไม่สามารถทำงานได้
Windows XP SP2	ไม่มี NX Bit	ไม่สามารถทำงานได้
	รองรับ NX Bit	ไม่สามารถทำงานได้

ตารางที่ 7 สรุปผลการทดลองกรณีโปรแกรมทดสอบ NX Bit

ระบบปฏิบัติการ	ประเภทของซีพียู	การตั้งค่าไฟล์ Boot.ini	ผลการทดลอง
Windows XP SP1	ไม่มี NX Bit	-	ทำงานได้ตามปกติ
Windows XP SP1	รองรับ NX Bit	-	ทำงานได้ตามปกติ

ตารางที่ 7 (ต่อ)

ระบบปฏิบัติการ	ประเภทของซีพียู	การตั้งค่าไฟล์ Boot.ini	ผลการทดลอง
Windows XP SP2	ไม่มี NX Bit	-	ทำงานได้ตามปกติ
Windows XP SP2	รองรับ NX Bit	/noexecute= OptIn	ทำงานได้ตามปกติ
		/noexecute= OptOut	ไม่สามารถทำงานได้

5. สรุปผลและข้อเสนอแนะ

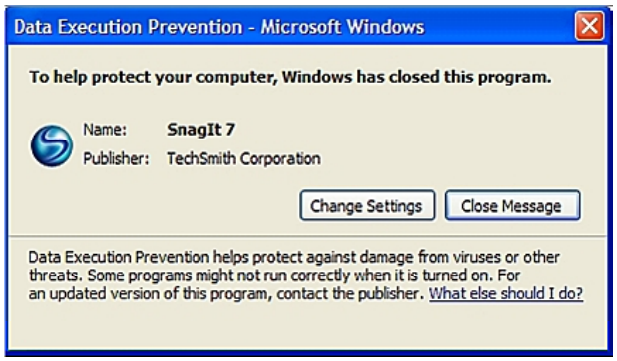
5.1 สรุปผล

โดยพื้นฐานของ Windows XP SP1 สามารถป้องกันระบบได้กรณีที่โปรแกรมเขียนขึ้นโดยจงใจให้เกิดปัญหา Overflow เพื่อความปลอดภัยของผู้ใช้งานควรอัพเดตขึ้นมาเป็น SP2 เพราะช่วยตัดการทำงานของโปรแกรมประเภทส่งแพ็กเก็ตรบกวนผ่านทางเครือข่ายหรือพวกสปายแวร์ได้บ้าง

การทำงานของ NX Bit สามารถหยุดโปรแกรมที่เข้าถึงตำแหน่งหน่วยความจำโดยตรง โปรแกรมพวกนี้ไม่ได้โจมตีผ่านทางเครือข่ายเพื่อเกิด Buffer Overflow โดยตรง แต่ปัญหาที่แท้จริงคือเมื่อเกิด Overflow ในระบบ ผู้ไม่ประสงค์ดีมักจะส่งโค้ดหรือโปรแกรมลับเข้ามาทำงานในคอมพิวเตอร์โดยผู้ใช้ไม่รู้ตัว โปรแกรมลับเหล่านี้ใช้เทคนิคเพื่อเจาะเข้าไปยังตำแหน่งหน่วยความจำโดยตรง NX Bit ช่วยลดจุดอ่อนของปัญหานี้

5.2 ข้อจำกัด

NX Bit จะเกิดประโยชน์เต็มที่ผู้ใช้ต้องมีคอมพิวเตอร์ที่ซีพียูรองรับ NX Bit และใช้งานบน Windows XP SP2 และต้องให้มีผลกับทุกโปรแกรม คือต้องกำหนดค่าในไฟล์ Boot.ini เป็น /noexecute =OptOut



ภาพที่ 9 NX Bit ยังพบปัญหากับโปรแกรมในปัจจุบัน

เมื่อกำหนดให้ NX Bit มีผลกับทุกโปรแกรมจะส่งผลกับบางโปรแกรมที่มีการเข้าถึงหน่วยความจำโดยตรง โดย NX Bit หยุดการทำงานโปรแกรมทันที เช่น SnagIt หรือ DivX Encoder เป็นต้น กรณีโปรแกรม DivX Encoder ปัจจุบันผู้ผลิตได้เปิดตัวรุ่น 5.2 ที่แก้ไขปัญหาแล้ว แต่ SnagIt ยังไม่มีการแก้ไข

5.3 ข้อเสนอแนะ

เมื่อ Windows Vista เปิดตัวอย่างเป็นทางการ อาจลองทดสอบบนระบบที่เป็น 64 บิตสมบูรณ์อีกครั้ง เพื่อเปรียบเทียบและดูผลการทำงานของระบบ ว่า NX Bit สามารถทำงานกับแอพพลิเคชัน 64 บิตได้สมบูรณ์ดังที่กล่าวอ้างหรือไม่

เอกสารอ้างอิง

[1] AMD. “AMD64 Architecture Programmer’s Manual Volume 2 : System Programming.” Advanced Micro Devices, Inc., Revision 3.10 February 2005.

[2] Mark E. Russinovich, David A. Solomon. “Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server 2003, Windows XP, and Windows 2000.” Microsoft Press, December 2004.