10

الفصل العاشر

البرمجيات الخبيثة

محتويات الفصل:

10-1 الفيروسات والتهديدات المتعلقة بها

10-2 الإجراءات المضادة للفيروسات

10-3 هجمات حجب الخدمة الموزَّعة

10-4 توصيات للمطالعة

10-5 مصادر للمعلومات على الويب

10-6 مصطلحات رئيسة

10-7 أسئلة للمراجعة ومسائل

"ما هو مفهوم الدفاع؟ تفادي ضربة. ما الخاصية المميزة له؟ انتظار الضربة."
ـ من كتاب "عن الحرب" ، لكارل فون كلوزويز.

النقاط الرئيسة

- البرمجيات الخبيثة هي البرامج التي تُضمَّن أو تُدخَل عمداً في نظامٍ ما لغرض ضار.
- الفيروس هو قطعة من البرمجيات التي يمكن أن تصيب البرامج الأخرى عن طريق تعديلها؛ يشمل التعديل إدخال نسخة من برنامج الفيروس التي يمكن بدورها أن تصيب برامج أخرى.
- الدودة هي برنامج يمكن أن يتكاثر (يضاعف نفسه) ويرسل نسخاً منه من حاسب إلى آخر عبر توصيلات الشبكة. قد تنشط الدودة عند وصولها للحاسب لمضاعفة نفسها وإعادة الانتشار من جديد. بالإضافة إلى الانتشار، عادةً ما تقوم الدودة ببعض العمليات غير المرغوبة.
- حجب الخدمة (DoS) هو هجوم يهدف إلى محاولة منع مستخدمين شرعيين لخدمة معينة من استخدام تلك الخدمة.
- يُشن هجوم حجب الخدمة الموزَّع من عدة مصادر متعاونة فيما بينها.

سنناقش في هذا الفصل البرمجيات الخبيثة (malware)، وخصوصاً الفيروسات والديدان.

10-1 الفيروسات والتهديدات المتعلقة بها

لعل أكثر أنواع التهديدات لأنظمة الحاسب تطوراً هذه الأيام يتمثل في البرامج التي تستغل نقاط الضعف في أنظمة الحاسب. ويشمل هذا السياق كلاً من برامج التطبيقات والبرامج الخدمية كبرامج التحرير (editors) والمترجمات

(compilers). وسنبدأ هذا الجزء بنظرة عامة على مدى التهديد الذي تمثله البرمجيات الخبيثة. وسنركز في باقي الجزء على الفيروسات والديدان.

10-1-1 البرمجيات الخبيثة

تُشكِّل المصطلحات في هذا المجال مشكلة نظراً لعدم وجود اتفاق عام على كل المصطلحات، ولتداخل بعض التصنيفات. يعطي الجدول 10-1 (والمستمد من [SZOR05]) دليلاً مفيداً لمصطلحات البرمجيات الخبيثة.

يمكن تقسيم البرمجيات الخبيثة إلى صنفين: صنف يحتاج إلى برنامج مضيف، وآخر مستقل. الصنف الأول هو في الحقيقة أجزاء برامج (fragments of programs) لا يمكنها التواجد بشكل مستقل عن برنامج تطبيق فعلي معين، أو برنامج خدمي أو برنامج نظام. ومن أمثلة تلك البرمجيات الخبيثة: الفيروسات (viruses)، والقنابل المنطقية (logic bombs)، والمداخل الخلفية (backdoors). الصنف الآخر هو برامج مكتفية ذاتياً ويمكن جدولتها (scheduling) وتشغيلها من قِبَل نظام التشغيل. من أمثلة ذلك: برامج الزومبي (zombies) والديدان (worms).

يمكن أيضاً أن نُفرِّق بين تهديدات البرامج التي لا تتكاثر (أي تتضاعف بنسخ نفسها) وتلك التي تتكاثر. النوع الأول هو برامج أو أجزاء برامج تحتاج لمثير لكي تنشط؛ كالقنابل المنطقية، والمداخل الخلفية، وبرامج الزومبي. أما النوع الآخر فيشمل أجزاء برامج أو برامج مستقلة قد تُنتج عند تنفيذها نسخة أو أكثر من نفسها لكي تنشط لاحقاً على نفس النظام أو على نظام آخر؛ كالفيروسات والديدان.

في بقيّة هذا الجزء الفرعي، سنستعرض بسرعة بعض تلك الأصناف الرئيسة من البرمجيات الخبيثة باستثناء الفيروسات والديدان والتي سوف نغطيها بتفصيل أكثر لاحقاً في هذا الجزء.

الجدول 10-1: مصطلحات البرمجيات الخبيثة.

المصطلح	الوصف
الفيروس (Virus)	يُلحق نفسه ببرنامج وينشر نسخاً من نفسه للبرامج الأخرى.
الدودة (Worm)	برنامج ينشر نسخاً من نفسه (ينتشر) من حاسب لآخر.
القنبلة المنطقية (Logic bomb)	تُثار عند توفر ظروف معينة.
حصان طروادة (Trojan horse)	برنامج يتضمن وظائف أخرى غير متوقعة.
مدخل خلفي (باب المصيدة) (Backdoor (trapdoor))	تعديل بالبرنامج يسمح بالوصول غير المخوّل للوظائف.
استغلالات (Exploits)	كود محدد بنقطة ضعف وحيدة أو مجموعة من نقاط الضعف.
برامج تنزيل (Downloaders)	برامج تقوم بتثبيت عناصر أخرى على الجهاز محل الهجوم. تُرسَل عادةً عبر البريد الإلكتروني.
برنامج للوصول التلقائي للجذر (Auto-rooter)	أدوات للمهاجم تُستخدم في اختراق أجهزة جديدة عن بُعد.
علبة أدوات (لتوليد فيروسات) (Kit (virus generator))	مجموعة من الأدوات لتوليد فيروسات جديدة بشكل آلي.
برامج إرسال رسائل الدعاية (Spammer programs)	تُستخدم لإرسال كم كبير من رسائل االبريد الإلكتروني غير المرغوبة.
برامج الفيضان (Flooders)	تُستخدم لمهاجمة أنظمة حاسب متصلة بالشبكة بكم هائل من حركة مرور البيانات لتنفيذ هجوم حجب الخدمة (DoS).
مسجلات ضربات المفاتيح (Keyloggers)	تُستخدم لتسجيل الضربات على لوحة المفاتيح بجهاز مصاب.
الجذور الخفية (Rootkit)	مجموعة من الأدوات التي يستخدمها المهاجم بعد أن يكون قد تَمكّن من اختراق نظام حاسب وكَسَب وصولاً للمستوى الجذري (root-level access) (أي مستوى مدير النظام ومن ثَمَّ يكون له صلاحيات أكثر).
برنامج الزومبي (Zombie)	برنامج يُفعَّل على جهاز مصاب لشن هجوم على أجهزة أخرى.

❖ المداخل الخلفية (Backdoors):

يُعرَف المدخل الخلفي أيضاً بباب المصيدة (trapdoor)، وهو عبارة عن نقطة دخول سرية للبرنامج يمكن استغلالها من قِبَل شخصٍ ما – على دارية بها – للدخول بدون المرور عبر الإجراءات الأمنية المعتادة للوصول. استخدم المبرمجون المداخل الخلفية بشكل شرعي لعدة سنوات من أجل تنقيح البرامج واختبارها. ويتم ذلك عادةً عندما يطور المبرمج تطبيقاً له إجراء توثيق، أو تهيئة طويلة، تتطلب من المستخدم إدخال عدد من القيم المختلفة لتشغيل التطبيق. ولإجراء عملية تنقيح (debugging) للبرنامج، قد يرغب المُبرمج في كسب امتيازات خاصة أو تفادي إجراءات التوثيق أو التهيئة. وقد يرغب المبرمج أيضاً في ضمان وجود طريقة لتنشيط البرنامج في حالة حدوث خطأ ما في إجراء التوثيق المُبيَّت (built-in) كجزء من التطبيق. والمدخل الخلفي هو تعليمات بالبرنامج تتعرف على تسلسل معيّن من المدخلات أو تُثار بتشغيلها بواسطة اسم مستخدم معين أو عبر تسلسل معيّن غير محتمل من الأحداث.

تُشكِّل المداخل الخلفية تهديدات أمنية عندما يستخدمها مبرمجون عديمو الضمير لكسب وصول غير مُخوَّل لهم. وكان المدخل الخلفي الفكرة الأساسية لنقطة الضعف التي جُسِّدت في فيلم "مناورات حربية" (War Games). من الأمثلة الأخرى أنه أثناء تطوير نظام التشغيل مولتكس (Multics)، أُجريت اختبارات الاختراق بواسطة "فريق النمر" التابع للقوّات الجوية (لمحاكاة الخصوم).

تضمَّنت إحدى الخطط (التكتيكات) التي استُخدِمت إرسال رسالة مزيفة لتحديث نظام التشغيل إلى موقع يستخدم نظام مولتكس. وتضمَّنت رسالة التحديث حصان طروادة – سنصفه لاحقاً – يمكن تنشيطه من قِبَل مدخل خلفي يسمح لفريق النمر بالتمكن من الدخول. ونُفِّذ التهديد بشكلٍ جيد جداً، لدرجة أن مُطوِّري نظام مولتكس أنفسهم لم يتمكنوا من العثور عليه، حتى بعد علمهم بوجوده [ENGE80].

من الصعب تنفيذ أساليب تحكم في المداخل الخلفية بنظام التشغيل. يجب أن تركّز إجراءات الأمن على عملية تطوير البرنامج والأنشطة المختلفة لتحديث البرمجيات.

❖ القنبلة المنطقية (Logic Bomb):

تُعدُّ القنبلة المنطقية أحد أقدم أنواع تهديدات البرامج، فهي تسبق الفيروسات والديدان. إن القنبلة المنطقية هي تعليمات مُتضمَّنة في برنامج شرعي، ومحدد لها وقت "انفجار" عندما تتحقق ظروف معينة. ومن أمثلة تلك الظروف التي يمكن أن تستخدم كمثيرات لانفجار القنبلة المنطقية: وجود بعض الملفات أو عدم وجودها، أو حلول يوم معيّن من الأسبوع أو تاريخ معين، أو تشغيل التطبيق من قِبَل مستخدم معين. عند إثارة القنبلة قد تقوم بتعديل أو حذف بيانات أو ملفات بكاملها، أو تتسبب في توقّف الجهاز، أو إلحاق أي ضرر آخر. من الأمثلة المثيرة على كيفية استخدام القنابل المنطقية قضية تيم لويد (Tim Lloyd) التي أُدين فيها بتثبيت قنبلة منطقية كلَّفت صاحب العمل (شركة أوميجا للهندسة) أكثر من 10 مليون دولار، وأحبطت الخطة الاستراتيجية لتطوير الشركة، وأدَّت إلى تسريح 80 من العاملين بالشركة [GAUD00]. في النهاية حُكِم على لويد بـ 41 شهراً في السجن وبدفع 2 مليون دولار تعويضات.

❖ أحصنة طروادة (Trojan Horses):

حصان طروادة هو برنامج أو إجراء مفيد (أو مفيد في ظاهره) يتضمن تعليمات مخفية تقوم عند تنفيذها ببعض الأعمال غير المرغوبة أو الضارّة.

يمكن أن تُستخدَم برامج أحصنة طروادة للقيام بشكلٍ غير مباشر بعمليات لا يمكن لمستخدم غير مخوّل له أن ينجزها بشكلٍ مباشر. فعلى سبيل المثال، للوصول إلى ملفات مستخدم آخر على نظام مشترك، يمكن أن ينشئ

المستخدم برنامج حصان طروادة يغيّر عند تنفيذه صلاحيات الوصول للملفات والمخوَّلة للمستخدم الذي يقوم بتنفيذ البرنامج بحيث يمكن لأي مستخدم قراءة تلك الملفات. ويمكن لمنشئ حصان طروادة أن يغري المستخدمين الآخرين بتشغيل البرنامج وذلك بوضعه في مجلد مُشترك (بنظام الملفات) ويسمّيه بحيث يبدو برنامجاً مفيداً. ومثال لذلك برنامج يزعم أنه يسرد ملفات المستخدم بصيغة مرغوبة. بعد أن يُشغّل مستخدم آخر ذلك البرنامج الذي يتضمن حصان طروادة، يمكن لمؤلف البرنامج عندئذ الوصول للمعلومات الموجودة في ملفات ذلك المستخدم. ومن الأمثلة لبرنامج حصان طروادة يصعب اكتشافه برنامج مترجم (compiler) مُعدَّل لإدخال تعليمات إضافية إلى بعض البرامج، كبرنامج الدخول على النظام، أثناء القيام بترجمتها [THOM84]. تنشئ تلك التعليمات مدخلاً خلفياً في برنامج الدخول على النظام مما يسمح لمؤلف تلك التعليمات بالدخول للنظام باستخدام كلمة سر خاصة. ولا يمكن أبداً اكتشاف حصان طروادة هذا بقراءة تعليمات كود المصدر (source code) لبرنامج الدخول.

يُعدُّ تدمير البيانات من الدوافع الأخرى الشائعة لاستخدام حصان طروادة. يظهر البرنامج كما لو كان يؤدّي وظيفة مفيدة (مثلاً برنامج آلة حاسبة)، لكنّه قد يقوم أيضاً خلسةً وبهدوء بحذف ملفات المستخدم. على سبيل المثال كان المدير التنفيذي لشركة CBS ضحيةً لحصان طروادة دمّر كل المعلومات الموجودة بذاكرة حاسبه [TIME90]. كان حصان طروادة هذا مزروعاً في إجراء رسومات (graphics routine) معروضٍ على لوحة إعلانات إلكترونية (electronic bulletin board).

❖ برنامج زومبي (Zombie):

هو برنامج يسيطر خلسةً على حاسب آخر متصل بالإنترنت، وبعد ذلك يستخدم ذلك الحاسب لشنّ هجمات يصعب تتبّعها إلى مُنشئ الزومبي. يُستخدم الزومبي في هجمات حجب الخدمة (DoS)، عادةً ضد مواقع ويب مستهدفة. يتم

زراعة الزومبي على مئات الحاسبات التي تنتمي لجهات أخرى لا تكون محل اشتباه، وبعد ذلك يُستخدَم البرنامج لغمر موقع الويب المستهدف بحركة مرور شديدة عبر الإنترنت. يناقش الجزء 10-3 الزومبي ضمن سياق هجمات حجب الخدمة.

10-1-2 طبيعة الفيروسات

الفيروس هو جزء من برنامج يمكن أن "يصيب" برامج أخرى بتعديلها؛ يشمل التعديل تضمين نسخة من برنامج الفيروس يمكن بدورها أن تواصل عملها بإصابة برامج أخرى.

الفيروسات الحيوية (biological viruses) (أي التي تصيب الكائن الحي) هي فضلات صغيرة من الشفرة الجينية – DNA أو RNA – يمكن أن تسيطر على الوحدات الوظيفية لخلية حيّة وتخدعها لعمل آلاف النُسخ المطابقة للفيروس الأصلي. مثل نظيره الحيوي، يحمل فيروس الحاسب في تعليماته وصفةً لإنتاج نُسخ تامة من نفسه. وبشكلٍ اعتيادي يكون الفيروس مُتضمَّناً في برنامج على حاسب. بعد ذلك عندما يتصل الحاسب المصاب بجزء غير مصاب من البرمجيات تعبر نسخة جديدة من الفيروس إلى البرنامج الجديد. وهكذا، يمكن أن تنتشر العدوى من حاسب إلى حاسب من قِبَل المستخدمين الغافلين من خلال تبادل الأقراص أو إرسال البرامج إلى أحدهما الآخر عبر شبكة. في بيئة الشبكة، توفر القدرة للوصول للتطبيقات ولخدمات النظام على حاسبات أخرى مزرعة مثالية لانتشار الفيروس.

يمكن أن يعمل الفيروس أي شيء يمكن أن تعمله البرامج الأخرى. ويكمن الفرق الوحيد في أنه يربط نفسه ببرنامج آخر ويُنفَّذ خلسةً عند تشغيل البرنامج المضيف. وأثناء تنفيذ الفيروس يمكن أن يؤدي أي عملية كمحو الملفات والبرامج. وأثناء فترة العمر للفيروس الاعتيادي يمر بالمراحل الأربع الآتية:

- مرحلة الخمول (dormant phase): وفيها يكون الفيروس خاملاً إلى أن يتم تنشيطه في النهاية بحدثٍ ما؛ كحلول تاريخ معين، أو وجود برنامج أو ملف آخر، أو تجاوز سعة تخزين القرص حداً معيناً. ويلاحظ أنه ليست كل الفيروسات تمر بهذه المرحلة.
- مرحلة الانتشار (propagation phase): يضع الفيروس نسخة مماثلة من نفسه على البرامج الأخرى أو في مناطق معينة خاصة بنظام التشغيل على القرص. ويحتوي كل برنامج مصاب الآن على نسخة من الفيروس تدخل بدورها هي الأخرى مرحلة انتشار جديدة.
- مرحلة الإثارة (triggering phase): يُنشَّط الفيروس لأداء الوظيفة المطلوبة منه. وكما هو الحال مع مرحلة الخمول، يمكن أن تنشأ عملية الإثارة بسبب تشكيلة مختلفة من أحداث النظام، بما في ذلك عدد المرات التي قامت فيها تلك النسخة من الفيروس بنسخ نفسها.
- مرحلة التنفيذ (execution phase): وفيها تُؤدّى العملية المطلوبة. قد تكون تلك العملية غير مؤذية كعرض رسالة على الشاشة، أو قد تكون ضارة كتدمير البرامج وملفات البيانات.

تُؤدي معظم الفيروسات عملها بطريقة خاصة بنظام تشغيل معين، وفي بعض الحالات بطريقة خاصة أيضاً بعتاد أجهزة معين. ومن ثَمَّ فهي مُصمّمة لاستغلال نقاط الضعف الموجودة بأنظمة بعينها.

❖ تركيب الفيروس:

يمكن أن يُضمَّن الفيروس في بداية برنامج قابل للتنفيذ أو نهايته، كما يمكن أن يُضمَّن في أشكالٍ أخرى. يكمن المفتاح لعمل الفيروس في أنه عند استدعاء البرنامج المصاب يتم تنفيذ تعليمات الفيروس أولاً ثم بعد ذلك تنفيذ تعليمات البرنامج الأصلي.

يُبيّن الشكل 10-1 تصوراً عاماً جداً لتركيب الفيروس (مأخوذاً من [COHE94]). في هذه الحالة تم تضمين تعليمات الفيروس V في بداية برامج مصابة، وافترض أن نقطة الدخول للبرنامج عنداستدعائه هي السطر الأول من البرنامج.

```
program V:=
{    goto main;
     1234567;

     subroutine infect-executable :=
     {loop:
          file := get-random-executable-file;
          if(first-line-of-file = 1234567) then
               goto loop;
          else
               prepend V to file;
     }

     subroutine do-damage :=
     {whatever damage is to be done}

     subroutine trigger-pulled :=
     {return true if some condition holds}

     main:  main-program :=
     {
          infect-executable;
          if trigger-pulled then do-damage;
          goto  next;
     }

     next:
}
```

الشكل 10-1: فيروس بسيط.

يبدأ البرنامج المصاب بتعليمات الفيروس ويعمل على النحو الآتي: تسبب أول تعليمة القفز إلى برنامج الفيروس الأساسي. التعليمة الثانية هي علامة خاصة تُستخدَم من قِبَل الفيروس لتحديد ما إذا كان برنامج الضحية المحتمل مصاباً بهذا الفيروس بالفعل أم لا. وعند استدعاء البرنامج، ينتقل التحكم فوراً إلى برنامج الفيروس الأساسي. يفتش برنامج الفيروس أولاً عن ملفات قابلة للتنفيذ غير مصابة ويصيبها. بعد ذلك قد يقوم الفيروس بعملٍ ما، عادةً ما يكون ضاراً بالنظام. يمكن أن يُؤدَّي هذا العمل في كل مرة يُستدعى فيها البرنامج، أو يمكن أن يكون قنبلة منطقية تثار فقط في ظروف معينة. وأخيراً، يُحوِّل الفيروس التحكم إلى البرنامج الأصلي. وإذا كانت مرحلة العدوى سريعة إلى حدٍّ معقول، فمن غير المحتمل أن يلاحظ المستخدم أي اختلاف بين تنفيذ البرنامج المصاب وغير المصاب.

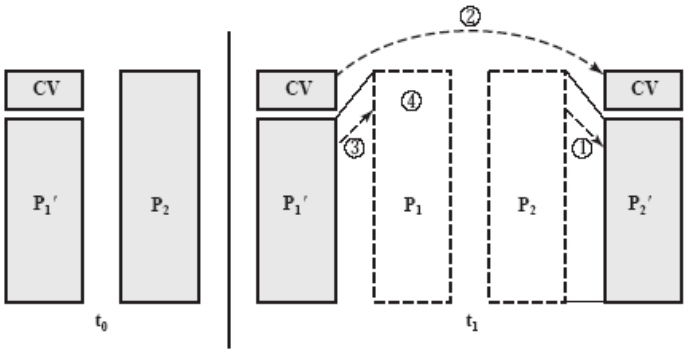
```
program CV:=
{    goto main;
     01234567;

     subroutine infect-executable :=
     {loop:
          file := get-random-executable-file;
          if(first-line-of-file = 01234567) then goto loop;
           (1)  compress file;
           (2)  prepend V to file;
     }

     main:  main-program :=
     {
          if ask-permission then infect-executable;
           (3)  uncompress rest-of-file;
           (4)  run uncompressed file;
     }
}
```

الشكل 10-2: منطق فيروس يقوم بضغط البرنامج الأصلي.

من السهل اكتشاف فيروس كالذي وصفناه للتو لأن حجم النسخة المصابة من البرنامج يكون أكبر من حجم النسخة المناظرة غير المصابة. ومن طرق إحباط مثل تلك الوسيلة البسيطة لاكتشاف الفيروس أن يقوم الفيروس بضغط الملف القابل للتنفيذ بحيث يتساوى حجم كلتا النسختين (المصابة وغير المصابة). يوضح الشكل 10-2 [COHE94] المنطق المطلوب بشكل عام؛ حيث تم ترقيم التعليمات المهمة في هذا الفيروس. يوضح الشكل 10-3 [COHE94] طريقة العمل.



الشكل 10-3: فيروس يقوم بضغط البرنامج الأصلي.

افترض أن البرنامج P_1 مصاب بالفيروس CV. عندما يُطلب تنفيذ ذلك البرنامج، ينتقل التحكم إلى الفيروس، والذي يقوم بالخطوات الآتية:

1. لكل ملف غير مصاب P_2 موجود، يضغط الفيروس أولاً ذلك الملف ليُنتج P'_2 بطول أقصر من البرنامج الأصلي بمقدار حجم الفيروس.
2. تُضاف نسخة من الفيروس في بداية البرنامج المضغوط.
3. يُزال ضغط النسخة المضغوطة P'_1 للبرنامج المصاب الأصلي.
4. يُنفَّذ البرنامج الأصلي غير المضغوط.

في هذا المثال، لا يعمل الفيروس شيئاً أكثر من الانتشار (العدوى). كما في المثال السابق، قد يتضمَّن الفيروس قنبلة منطقية.

❖ **العدوى الأولى:**

بمجرد أن يتمكن الفيروس من الدخول على النظام بإصابة برنامج واحد، يصبح قادراً على إصابة بعض الملفات الأخرى القابلة للتنفيذ أو كلها على ذلك النظام عند تنفيذ البرنامج المصاب. ومن ثَمَّ يمكن منع العدوى الفيروسية بالكامل بمنع الفيروس من الدخول في المقام الأول. ولسوء الحظ، عملية المنع تلك صعبة جداً لأن الفيروس يمكن أن يكون جزءاً من أي برنامج خارج النظام. ولذلك ما لم يكن الشخص مقتنعاً بأخذ قطعة خالية من الحديد ثم يكتب عليها كل ما يخصه من برامج النظام والتطبيقات، فسيكون في موقف ضعف.

10-1-3 أنواع الفيروسات

منذ ظهور الفيروسات والصراع مستمرٌ بين كُتَّاب برامج الفيروسات وكُتَّاب البرامج المضادة للفيروسات. وكلما طُوِّرت إجراءات مضادة فعَّالة لأنواع الفيروسات الموجودة، طُوِّرت أنواع جديدة أخرى من الفيروسات. واقترح [STEP93] الفئات الآتية لتمثيل أشد أنواع الفيروسات خطورة:

- الفيروس الطفيلي (parasitic virus): وهو النوع التقليدي الذي لا يزال يمثل الشكل الأكثر شيوعاً من الفيروسات. يربط الفيروس الطفيلي نفسه بالملفات القابلة للتنفيذ، ويتضاعف عندما يُنفَّذ البرنامج المصاب بالعثور على الملفات الأخرى القابلة للتنفيذ وإصابتها.
- الفيروس المقيم بالذاكرة (memory-resident virus): يسكن في الذاكرة الرئيسة كجزء من برنامج مقيم من برامج نظام التشغيل. ومن تلك اللحظة فصاعداً، يُصيب الفيروس كل البرامج التي تُنفَّذ.

- فيروس قطاع البدء (boot sector virus): يصيب سجل بدء رئيس (master boot record) أو سجل بدء (boot record) وينتشر عند بدأ تشغيل النظام من قرص يحتوي على الفيروس.
- الفيروس المُتخفي (stealth virus): شكل من أشكال الفيروسات مصمّم بطريقة تُمكّنه من إخفاء نفسه بحيث لا تتمكّن البرامج المضادة للفيروسات من اكتشافه.
- الفيروس متعدد الأشكال (polymorphic virus): فيروس يُغيّر شكله مع كل عدوى، مما يجعل عملية اكتشافه عن طريق "بصمته" مستحيلة.
- الفيروس المَسْخي (المتغير المظهر) (metamorphic virus): كما هو الحال مع الفيروس متعدد الأشكال، يتغيّر الفيروس المسخي في كل عدوى. الفرق بينهما هو أن الفيروس المسخي يعيد كتابة نفسه بالكامل في كل تكرار، مما يزيد من صعوبة اكتشافه. قد تُغيِّر الفيروسات المسخية من سلوكها بالإضافة إلى مظهرها.

تم مناقشة أحد الأمثلة للفيروسات المتخفية في وقت سابق: الفيروس الذي يستخدم الضغط (compression) لكي يكون حجم البرنامج المصاب هو نفسه بالضبط قبل الإصابة. هناك طرق ممكنة تُعدُّ أكثر تطوراً بكثير من ذلك. على سبيل المثال، يمكن أن يضع الفيروس "منطقاً اعتراضياً" (intercept logic) في برامج الدخل والخرج للأقراص (disk I/O routines)، بحيث إنه عند محاولة المستخدم قراءة الأجزاء المشكوك فيها بالقرص لاكتشاف الفيروس يقوم الفيروس بتقديم البرنامج الأصلي غير المصاب. وهكذا، فإن مصطلح "التخفي" ليس تعبيراً ينطبق على الفيروس في حد ذاته لكنه بالأحرى طريقة يستخدمها الفيروس في المراوغة وتجنب الاكتشاف.

أثناء عملية الانتشار، يُنشئ الفيروس متعدد الأشكال نسخاً متكافئة من حيث الوظيفة ولكن لها أنماط بتات مختلفة. كما هو الحال مع الفيروس المتخفي، الهدف من ذلك هو هزيمة البرامج التي تُستخدم لكشف الفيروسات. وفي هذه الحالة، يتغير "توقيع" الفيروس مع كل نسخة. ولتحقيق هذا الاختلاف، قد يُدخِل الفيروس تعليمات زائدة عن الحاجة بشكل عشوائي أو يبدل ترتيب التعليمات المستقلة بالبرنامج. من الطرق أخرى الأكثر فعالية استخدام التشفير. يُنشئ جزء من الفيروس، يطلق عليه بشكلٍ عام "مُحرِّك التغيير" (mutation engine)، مفتاح تشفير عشوائي لتشفير الجزء المتبقي من الفيروس. يُخزَّن المفتاح مع الفيروس، ويتم تعديل مُحرِّك التغيير نفسه. عندما يُطلب تنفيذ برنامج مصاب، يستخدم الفيروس المفتاح العشوائي المُخزَّن لإزالة تشفير الفيروس. وعندما يتكاثر الفيروس يختار مفتاحاً عشوائياً آخر.

من الأسلحة الأخرى في ترسانة كُتّاب الفيروسات استخدام حزمة برمجيات (toolkit) لإنشاء الفيروسات بشكل آلي. تُمكِّن مثل تلك البرمجيات الشخص المبتدئ نسبياً من إنشاء عدد من الفيروسات المختلفة بسرعة. ورغم أن الفيروسات التي تنتجها تلك البرمجيات تكون عادةً أقل تطوراً من الفيروسات الأخرى التي صُمِّمت من البداية بشكل مستقل، فإن العدد الهائل من الفيروسات الجديدة التي يمكن إنتاجها بهذه الطريقة يُمثّل مشكلة تواجه الأساليب المضادة للفيروسات.

10-1-4 فيروسات الماكرو (Macro Viruses)

في منتصف التسعينيات، أصبحت فيروسات الماكرو إلى حدٍّ كبير أكثر أنواع الفيروسات انتشاراً. تُشكل تلك الفيروسات على وجه الخصوص تهديداً أمنياً كبيراً لعدة أسباب:

- لا يعتمد فيروس الماكرو على منصَّة (platform) الأجهزة المستخدمة. فمثلاً كل فيروسات الماكرو تقريباً تصيب مستندات برنامج

مايكروسوفت لمعالجة الكلمات (Microsoft word). ومن ثَمَّ فإن أي عتاد أجهزة أو نظام تشغيل يدعم هذا البرنامج يمكن إصابته بتلك الفيروسات.

- تصيب فيروسات الماكرو المستندات، وليس الأجزاء القابلة للتنفيذ من البرامج. لاحظ أن أغلب المعلومات على نظام الحاسب توجد على شكل مستندات وليس على شكل برامج.
- تنتشر فيروسات الماكرو بسهولة. يمثل البريد الإلكتروني طريقة شائعة جداً لتحقيق ذلك.

تستغل فيروسات الماكرو خاصية موجودة ببرنامج معالجة الكلمات (word) وغيره من تطبيقات المكتب الأخرى كبرنامج مايكروسوفت إكسل للجداول الإلكترونية، ألا وهي الماكرو (macro). بشكلٍ أساسي، الماكرو هو برنامج قابل للتنفيذ مُضمَّن في مستند معالجة الكلمات أو نوع آخر من الملفات. عادةً ما يُستخدم الماكرو لأتمتة المهام المتكررة وذلك لتقليل حاجة مستخدمي التطبيق إلى الضغط على المفاتيح. وفي العادة تكون لغة البرمجة للماكرو نوعاً من لغة البيسك (BASIC). ويمكن أن يُعرِّف المستخدم سلسلة من ضربات المفاتيح في ماكرو ويُعدِّها بحيث يُستدعى الماكرو عند الضغط على أحد مفاتيح الوظائف (function keys) أو على مجموعة خاصة مختصرة من المفاتيح.

توفر الإصدارات المتعاقبة من برنامج معالجة الكلمات حمايةً متزايدةً ضد فيروسات الماكرو. فعلى سبيل المثال، توفر مايكروسوفت أداة اختيارية للحماية من فيروسات الماكرو تساعد على اكتشاف ملفات برنامج word المريبة، وتقوم بلفت نظر المستخدم إلى الخطر المحتمل عند فتح ملف يتضمن ماكرو. وطوَّرت أيضاً العديد من شركات البرامج المضادة للفيروسات منتجات للكشف عن فيروسات الماكرو وتصحيحها. كما هو الحال مع أنواع الفيروسات الأخرى، يستمر السباق في حقل فيروسات الماكرو، غير أنها لم تعد تمثل تهديد الفيروسات السائد.

5-1-10 فيروسات البريد الإلكتروني

يُعدُّ فيروس البريد الإلكتروني تطوراً أحدث في البرمجيات الخبيثة. استَخدَمت فيروسات البريد الإلكتروني الأولى التي انتشرت بسرعة، كفيروس ميليسا (Melissa)، ماكرو مُتضمِّناً في مستندات word الملحقة برسالة البريد. إذا فتح المُستلِم ملحق البريد الإلكتروني، فإنه يتم تنشيط الماكرو، ومن ثم:

1. يُرسل فيروس البريد الإلكتروني نفسه إلى كل شخص على قائمة العناوين في حزمة البريد الإلكتروني الخاصة بالمستخدم.
2. يتسبب الفيروس في أضرار محلية.

في نهاية عام 1999 ظهرت نوعيّة أكثر قوة من فيروس البريد الإلكتروني، حيث يمكن تنشيط تلك النوعية الأحدث بمجرد فتح رسالة البريد الإلكتروني التي تحتوي الفيروس دون الحاجة إلى فتح الملحق. يستخدم الفيروس لغة برمجة البيسك المرئية (Visual Basic) والمدعومة من قِبَل حزمة البريد الإلكتروني.

وهكذا نرى جيلاً جديداً من البرمجيات الخبيثة التي تصل عن طريق البريد الإلكتروني وتستخدم خصائص تطبيق البريد الإلكتروني للانتشار (مضاعفة نفسها) عبر الإنترنت. ينشر الفيروس نفسه بمجرد تنشيطه (بفتح البريد الإلكتروني نفسه أو بفتح أحد ملحقاته)، وينتقل لكل عناوين البريد الإلكتروني المعروفة لدى المضيف المصاب. ونتيجة لذلك أصبحت الفيروسات تنتشر الآن في غضون ساعات، بعد أن كان ذلك يستغرق شهوراً وسنين في السابق. لذا أصبح من الصعب جداً على البرامج المضادة للفيروسات الاستجابة لخطر تلك الفيروسات لتلافي حدوث ضرر كبير. وفي النهاية، ينبغي تزويد برامج الخدمات المرافقة للإنترنت وبرامج التطبيقات على الحاسب الشخصي بدرجة أكبر من الأمن لمواجهة ذلك التهديد المتزايد.

6-1-10 الديدان

الدودة هي برنامج يمكن أن يضاعف نفسه ويرسل النسخ من حاسب لآخر عبر توصيلات الشبكة. وعند وصولها، قد تنشط الدودة وتقوم بالتكاثر والانتشار مرة أخرى. بالإضافة إلى الانتشار، تؤدّي الدودة عادةً بعض الأعمال غير المرغوبة. لفيروس البريد الإلكتروني بعض خصائص الديدان، فهو ينتقل من نظام إلى آخر. ومع ذلك ما زلنا نصنّفه كفيروس لأنه يتطلب شخصاً لدفعه إلى الأمام. تنشط الدودة في البحث عن أجهزة أخرى لإصابتها، ويُستخدم كل جهازٍ مصابٍ بدوره كمنصَّة لشن هجمات على أجهزة أخرى.

تستخدم برامج ديدان الشبكة توصيلات الشبكة للانتشار من نظام لآخر. وبمجرد أن تنشط داخل نظامٍ ما، يمكن أن تتصرف دودة الشبكة كفيروس أو بكتيريا، كما يمكن أن تزرع أحصنة طروادة، أو أن تؤدي أي عددٍ من الأعمال المعرقلة أو التدميرية. ولمضاعفة نفسها تستخدم دودة الشبكة أحد وسائل الاتصال المتاحة عبر الشبكة؛ ومن أمثلة ذلك:

- البريد الإلكتروني: تُرسل الدودة نسخة من نفسها كرسالة بريد إلى الأنظمة الأخرى.
- إمكانية التنفيذ عن بُعد (remote execution): تُنفِّذ الدودة نسخة من نفسها على نظام آخر.
- إمكانية الدخول على الأنظمة عن بُعد (remote login): تُسجِّل الدودة للدخول على نظام بعيد كمُستخدِم ثم تصدر تعليمات لنسخ نفسها من نظامٍ إلى آخر.

تعمل النسخة الجديدة من برنامج الدودة على النظام البعيد حيث تقوم، بالإضافة إلى أي وظائف تؤديها على ذلك النظام، بمواصلة نشر نفسها بنفس الطريقة.

لدودة الشبكة نفس خصائص فيروس الحاسب: مرحلة الخمول، ومرحلة الانتشار، ومرحلة الإثارة، ومرحلة التنفيذ. بشكل عام، تتضمن مرحلة الانتشار الوظائف الآتية:

1. البحث عن أنظمة أخرى لإصابتها بفحص جداول المضيف (أو مستودعات مماثلة) التي تتضمن عناوين أنظمة بعيدة.
2. تنشئ اتصالاً بالنظام البعيد.
3. تنسخ نفسها للنظام البعيد وتتسبب في تشغيل تلك النسخة على ذلك النظام.

قد تحاول دودة الشبكة أيضاً تحديد ما إذا كان النظام قد سبق أن أصيب أم لا. في أنظمة البرمجة المتعدّدة، قد تقوم الدودة أيضاً بإخفاء تواجدها بتسمية نفسها كعملية نظام أو باسم آخر لا يمكن لمُشغِّل النظام ملاحظته. كما هو الحال مع الفيروسات، يصعب مواجهة ديدان الشبكة.

❖ دودة موريس:

حتى وقت ظهور الجيل الحالي من الديدان، كانت الدودة الأكثر شهرة هي تلك التي قام روبرت موريس بإطلاقها إلى الإنترنت في عام 1998. كانت دودة موريس مُصمَّمة للانتشار على أنظمة يونيكس واستخدمت عدداً من الطرق المختلفة للانتشار. وعندما تبدأ نسخة في التنفيذ، تكون مهمتها الأولى اكتشاف المضيفات الأخرى المعروفة لهذا المضيف التي تسمح بالدخول عليها من ذلك المضيف. تقوم الدودة بذلك عن طريق فحص تشكيلة من القوائم والجداول، بما في ذلك جداول النظام التي تحدد الأجهزة الأخرى التي تُعتبر محل ثقة لدى ذلك المضيف، وملفات تمرير البريد الإلكتروني، والجداول التي من خلالها يُعطَى المستخدمون صلاحيات الوصول إلى الحسابات عن بُعد، ومن ثم إنشاء برنامج يقوم بالإبلاغ عن حالة توصيلات الشبكة.

لكل مضيف يتم اكتشافه، تُجرِّب الدودة عدة طرقٍ مختلفةٍ لمحاولة الدخول:

1. تحاول التسجيل للدخول على المضيف البعيد كمستخدم شرعي. في هذه الطريقة تحاول الدودة في البداية اختراق ملف كلمة السر المحلي، وبعد ذلك تستخدم الأزواج المكتشفة لكلمات السر وهويات المستخدمين المناظرة. وتكمن الفرضية في أن عدداً كبيراً من المستخدمين يستخدمون نفس كلمة السر على نظم مختلفة. وللحصول على كلمات السر، تُشغِّل الدودة برنامجاً لكسر كلمات السر؛ يجرب البرنامج:
   a. كل اسم حساب للمستخدم والتبديلات البسيطة منه،
   b. قائمة من 432 كلمة سر مبيّتة (built-in) داخل دودة موريس اعتقد موريس أنها محتملة بدرجة كبيرة.
   c. كل الكلمات في دليل النظام المحلي.
2. تستغل خطأً برمجياً في بروتوكول finger للإبلاغ عن مكان المستخدم البعيد.
3. تستغل مدخلاً خلفياً في خيار التنقيح للعملية البعيدة التي تستلم وترسل البريد.

إذا نجحت أيٌّ من تلك الهجمات، تكون الدودة قد تمكنت من تحقيق اتصال بمُفسِّر الأوامر (command interpreter) لنظام التشغيل. ومن ثَمَّ تُرسل لهذا المُفسِّر برنامج بدء ذاتي (bootstrap) قصير، وتُصدر أمراً لتنفيذ ذلك البرنامج، وبعدها تُسجِّل خروجاً من النظام. بعد ذلك يستدعى برنامج البدء الذاتي البرنامج الأصلي ويُنزّل بقيّة الدودة، وبعدها تُنفَّذ الدودة الجديدة.

❖ هجمات الديدان المعاصرة:

بدأ العصر الحديث لتهديدات الدودة مع إطلاق دودة الكود الأحمر (Red Code) في يوليو/تموز من عام 2001. تستغل دودة الكود الأحمر ثغرةً أمنيةً في خادم معلومات الإنترنت ((Internet Information Server (IIS) من مايكروسوفت لاختراق النظام ومن ثم الانتشار من خلاله. تقوم الدودة أيضاً بتعطيل مدقّق ملفات النظام (system file checker) في نظام ويندوز. وتتلمّس الدودة كذلك عناوين IP عشوائية للانتشار إلى مضيفات أخرى. وخلال فترة زمنية معينة، تقتصر الدودة على الانتشار فقط. بعد ذلك تبدأ بشن هجوم حجب الخدمة ضد موقع ويب حكومي، وذلك بإرسال فيضان من الرزم إلى الموقع من عدة مضيفات. ثم تُعلِّق الدودة أنشطتها لتعاود نشاطها من جديد بعد ذلك بشكل دوري. في موجة الهجوم الثانية، أصابت دودة الكود الأحمر ما يقرب من 360 ألف خادم في غضون 14 ساعة. بالإضافة إلى الدمار الذي سببته في الخادمات المستهدفة، يمكن أن تستهلك الدودة حيزاً كبيراً من سعة الإرسال على الإنترنت مما يعرقل الخدمة.

كان الإصدار الثاني للدودة، الكود الأحمر 2 (Code Red II)، نسخة جديدة تستهدف خادمات مايكروسوفت للويب (IIS). إضافةً إلى ذلك، تُثبّت هذه الدودة الحديثة مدخلاً خلفياً يسمح لمهاجمي الحاسبات بتوجيه أنشطة الحاسبات الضحيّة. وفي أواخر عام 2001 ظهرت دودة أكثر مرونة، عُرِفت باسم نيمدا (Nimeda). انتشرت نيمدا بآليات متعدّدة:

- من زبون لآخر عن طريق البريد الإلكتروني.
- من زبون لآخر عن طريق موارد الشبكة المشتركة المفتوحة.
- من خادم ويب إلى زبون عن طريق تصفح مواقع الويب المصابة.

- من زبون إلى خادم ويب عن طريق المسح النشط واستغلال نقاط الضعف المختلفة المرتبطة باستعراض دليل خادم مايكروسوفت للويب الإصدار 4.0 و5.0.
- من زبون إلى خادم ويب عن طريق المسح بحثاً عن المداخل الخلفية التي تنشئها ديدان "الكود الأحمر 2".

تُعدِّل الدودة مستندات الويب (كملفات htm، html، asp) وملفات معينة قابلة للتنفيذ موجودة على الأنظمة التي تصيبها، وتُنشئ نسخاً عديدة من نفسها بأسماء ملفات مختلفة. وفي أوائل عام 2003 ظهرت دودة إس كيو إل سلامار (SQL Slammer). استغلت تلك الدودة نقطة الضعف المتعلقة بالفيضان (overflow) في خادم مايكروسوفت لقواعد بيانات SQL. وكانت سلامار مختصرة جداً وانتشرت بسرعة، مما تسبب في إصابة 90% من المضيفات الضعيفة خلال 10 دقائق. وشهدت أواخر عام 2003 ظهور دودة سوبيج.إف (Sobig.f) التي استغلت خادمات البروكسي المفتوحة لتحويل الأجهزة المصابة إلى محركات رسائل الدعاية (spam engines). وفي ذروة نشاطها كانت دودة سوبيج.إف (Sobig.f) مسؤولة عن توليد رسالة واحدة من بين كل 17 رسالة وأنتجت أكثر من مليون نسخة من نفسها خلال الساعات الأربع والعشرين الأولى من إطلاقها.

دودة مايدوم (Mydoom) هي دودة لرسائل البريد الإلكتروني تُرسَل لعدد كبير من المستخدمين، وظهرت في عام 2004. واتّبع مصممو تلك الدودة الاتجاه المتزايد لتثبيت مداخل خلفية في الحاسبات المصابة، مما يُمكِّن مهاجمي الحاسبات من الوصول لبيانات مهمة عن بُعد، ككلمات السر وأرقام بطاقات الائتمان. تكاثرت دودة "مايدوم" بمعدل ألف مرة في الدقيقة وأغرقت الإنترنت بحوالي 100 مليون رسالة مصابة في 36 ساعة.

7-1-10 الوضع الحالي لتقنية الديدان

يشمل الوضع الحالي لتقنية الديدان ما يأتي:

- تعدد المنصَّات (multiplatform): لا تقتصر الديدان الحديثة على إصابة أجهزة الحاسب التي تعمل بنظام ويندوز فقط، بل يمكن أن تهاجم منصَّات مختلفة، خاصةً تشكيلة أنظمة يونيكس شائعة الاستخدام.
- تعدد الاستغلال (multiexploit): تخترق الديدان الجديدة الأنظمة بتشكيلة متنوعة من الطرق باستخدام طرق مختلفة من الاستغلال (exploits) ضد نقاط الضعف في خادمات الويب، والمتصفحات، والبريد الإلكتروني، والملفات المشتركة، وتطبيقات الشبكة الأخرى.
- الانتشار السريع جداً (ultrafast spreading): واحد من الطرق المتّبعة لتسريع انتشار الدودة، حيث يتم إجراء مسح مُسبَق للإنترنت لتجميع عناوين الإنترنت للأجهزة الضعيفة لاستهدافها.
- تعدد الأشكال (polymorphic): لتجنب الاكتشاف والمرور عبر المرشحات وإحباط محاولات التحليل الفوري (الوقت الحقيقي) (real-time analysis)، تتبع الديدان أسلوب تعدد الأشكال الذي تستخدمه الفيروسات. حيث يكون لكل نسخة من الدودة تعليمات جديدة تنشئ في التوّ باستخدام تعليمات وأساليب تشفير متكافئة وظيفياً.
- الديدان المسخية (متعددة المظهر) (metamorphic): بالإضافة إلى تغيير مظهرها، للديدان المسخية عدة أنماط من السلوك التي يتم إطلاقها في المراحل المختلفة من عملية الانتشار.
- الديدان كوسائل نقل (transport vehicles): نظراً لأن الديدان يمكن أن تصيب عدداً كبيراً من الأنظمة بسرعة، فإنها تُعدُّ طريقة مثالية

لنشر أدوات أخرى للهجمات الموزَّعة، كهجوم زومبي الموزَّع لحجب الخدمة.

- استغلال ساعة الصفر (zero-day exploit): لتحقيق أكبر قدرٍ من المفاجأة وأقصى توزيع ممكن، يجب أن تستغل الدودة نقطة الضعف المجهولة التي اكتشفها العاملون في مجال الشبكات العامة فقط عند إطلاق الدودة.

10-2 الإجراءات المضادة للفيروسات

10-2-1 الأساليب المضادة للفيروسات

الحل المثالي لتهديدات الفيروسات هو الوقاية منها؛ أي عدم السماح للفيروس بالدخول للنظام في المقام الأول. تحقيق هذا الهدف عموماً مستحيل، رغم أن المنع يمكن أن يُخفّض عدد الهجمات الفيروسية الناجحة. الطريقة الأفضل بعد الوقاية أن تكون قادراً على عمل الآتي:

- الكشف (detection): عندما تحدث العدوى، يتضمن الكشف التأكد من أن العدوى حدثت وتحديد موضع الفيروس.
- تحديد الهوية (identification): بمجرد إنجاز الاكتشاف، يلزم التعرف على نوع الفيروس الذي أصاب البرنامج.
- الإزالة (removal): بعد تحديد الفيروس، يلزم إزالة كل آثاره من البرنامج المصاب وإعادة البرنامج لحالته الأصلية. ينبغي إزالة الفيروس من كل الأنظمة المصابة لإيقاف انتشار العدوى.

إذا نجح الكشف عن الفيروس لكن لم يتم التعرف عليه أو لم يمكن إزالته، فإن البديل هو استبعاد البرنامج المصاب وإعادة تحميل نسخة نظيفة من النسخة الاحتياطية.

تسير التطورات في تقنية الفيروسات وتقنية البرامج المضادة لها يداً بيد. كانت الفيروسات الأولى بسيطة نسبياً، ويمكن تمييزها والتخلص منها بحِزَم برامج بسيطة مضادة للفيروسات. ومع استمرار تطور سباق التسلح ضد الفيروسات، ازدادت الفيروسات، وبالطبع البرامج المضادة لها، تعقيداً وتطوراً. حدد [STEP93] أربعة أجيال للبرامج المضادة للفيروسات:

- الجيل الأول: برامج مسح (scanners) بسيطة.
- الجيل الثاني: برامج مسح اجتهادية (heuristic scanners).
- الجيل الثالث: مصائد أنشطة (activity traps).
- الجيل الرابع: الحماية الكاملة.

يتطلب برنامج المسح من الجيل الأول معرفة توقيع (بصمة) الفيروس ليتمكن من تمييزه. قد يتضمن الفيروس رموزاً عامة (wildcards) لكنه أساساً له نفس التركيب ونمط البتات في كل نسخة. وتصلح مثل تلك البرامج المرتبطة بتوقيعات معينة للكشف عن الفيروسات المعروفة فقط. ويحتفظ نوع آخر من برامج الجيل الأول بسجل لأطوال البرامج ويبحث عن التغييرات في الطول.

أما برامج الجيل الثاني فلا تعتمد على توقيعات معينة، وإنما تستخدم قواعد اجتهادية (heuristic rules) للبحث عن عدوى الفيروس المحتملة. ويبحث أحد أنواع ذلك الصنف من البرامج عن أجزاء الكود (code fragments) المرتبطة غالباً بالفيروسات. فعلى سبيل المثال قد يبحث البرنامج عن بداية حلقة تشفير (encryption loop) تُستخدم مع الفيروسات متعددة الأشكال ويكتشف مفتاح التشفير. عندما يكتشف المفتاح، يمكن أن يُزيل برنامج الماسح التشفير للتعرّف على الفيروس، ثمّ يزيل العدوى ويعيد البرنامج للعمل.

من الطرق الأخرى التي يستخدمها الجيل الثاني فحص السلامة (integrity checking). يمكن أن يُذيّل كل برنامج بمجموع تدقيقي (checksum). إذا

أصاب الفيروس البرنامج بدون تغيير قيمة المجموع التدقيقي الملحقة بالبرنامج، فسيتمكن فحص السلامة من اكتشاف الفيروس. لمواجهة فيروس متطوّر بما فيه الكفاية لتغيير القيمة المُخزَّنة للمجموع التدقيقي عندما يصيب البرنامج، يمكن استخدام كود التحوير المُشفَّرة (encrypted hash function). يُخزّن مفتاح التشفير بشكلٍ منفصل عن البرنامج حتى لا يتمكن الفيروس من إنشاء كود تحوير جديد ويقوم بتشفيره. باستخدام كود التحوير بدلاً من المجموع التدقيقي البسيط، سيُمنع الفيروس من تعديل البرنامج ليعطي نفس كود التحوير كما في السابق.

برامج الجيل الثالث برامج مقيمة بالذاكرة تتعرّف على الفيروس من سلوكه بدلاً من تركيبه في برنامج مصاب. تمتاز تلك البرامج بأنها لا تحتاج لتطوير توقيعات وقواعد اجتهادية لعدد كبير من الفيروسات، وإنما من الضروري فقط تمييز مجموعة صغيرة من الأفعال التي تصاحب محاولة الإصابة بالعدوى ومن ثم التدخل.

منتجات الجيل الرابع هي حزم برمجيات تشمل تشكيلة من الأساليب المضادة للفيروسات والتي تُستخدم معاً، بما في ذلك المسح ومصائد الأنشطة. بالإضافة إلى ذلك، تتضمّن الحزمة إمكانية التحكم في الوصول التي تُحِد من قدرة الفيروسات على اختراق النظام ومن ثَمَّ تُحِد من قدرتها على تعديل الملفات لنقل العدوى.

يستمر سباق التسلح ضد الفيروسات. وتستخدم حزم الجيل الرابع استراتيجيةً أكثر شمولية للدفاع، مما سيوسّع مجال الدفاع ليشمل إجراءات أكثر عمومية لأمن الحاسب.

10-2-2 الأساليب المتقدمة المضادة للفيروسات

يستمر ظهور أساليب ومنتجات أكثر تطوّراً لمكافحة للفيروسات. سنُلقي هنا الضوء على أسلوبين مهمين منها.

❖ **طريقة إزالة التشفير العامة:**

تُمكِّن طريقة إزالة التشفير العامة ((Generic decryption (GD) البرامج المضادة للفيروسات من اكتشاف الفيروسات بسهولة، حتى الفيروسات المتطورة متعددة الأشكال، مع الاحتفاظ بسرعة مسح عالية [NACH97]. تذكَّر أنه عند تنفيذ ملف يحتوي على فيروس متعدد الأشكال، يجب أن يُزيل الفيروس تشفير نفسه لينشط. لكي يُكتشَف مثل هذا التركيب، يجب تشغيل الملفات القابلة للتنفيذ من خلال ماسح GD يتضمن العناصر الآتية:

- مُحاكٍ لوحدة المعالجة المركزية: وهو برنامج يمثل حاسباً افتراضياً. تفسّر التعليمات الموجودة في ملف تنفيذي بالمُحاكي بدلاً من تنفيذها على المعالج التحتي. ويتضمّن المُحاكي نُسخاً برمجية من كل المسجِّلات (registers) والمكونات المادية الأخرى للمعالج حتى لا يتأثر المعالج التحتي بالبرامج التي تفسر بواسطة المُحاكي.
- ماسح لتوقيعات الفيروسات: وحدة لمسح الكود المستهدف بحثاً عن توقيعات الفيروسات المعروفة.
- وحدة تحكم في المحاكاة: وتتحكم في تنفيذ الكود المستهدف.

في بداية كل محاكاة، يبدأ المُحاكي بترجمة التعليمات في الكود المستهدف، الواحدة تلو الأخرى. ومن ثَمَّ إذا تضمَّن الكود إجراءً لإزالة التشفير وكشف الفيروس، فإن ذلك الكود سيُترجم. فعلياً، يقوم الفيروس بالعمل لحساب البرنامج المضاد للفيروسات وذلك بالإعلان عن الفيروس وكشفه.

بشكل دوري، تقوم وحدة التحكم بمقاطعة عملية التفسير لمسح الكود المستهدف بحثاً عن توقيعات الفيروسات.

لا يمكن أن تُلحِق عملية تفسير الكود المستهدف أي ضرر ببيئة الحاسب الشخصي الفعلية، حيث تتم الترجمة بالكامل في بيئة تحت السيطرة.

يُعدُّ تحديد مدة التشغيل لكل تفسير أصعب قضايا تصميم ماسح GD. عادةً ما تُنشَّط عناصر الفيروس بعد بدء تنفيذ البرنامج مباشرةً، ولكن ليس من الضروري أن يكون الأمر كذلك. كلما طالت الفترة التي يقوم فيها الماسح بمحاكاة برنامج معين، زاد احتمال اكتشاف أي فيروسات مخفية. ومع ذلك، يمكن أن يستغرق البرنامج المضاد للفيروسات مدة محدودة ويستهلك موارد محدودة قبل أن يشتكي المستخدمون.

❖ نظام المناعة الرقمي (Digital Immune System):

نظام المناعة الرقمي هو طريقة شاملة للحماية ضد الفيروسات قامت بتطويره شركة IBM [KEPH97a, KEPH97b]، وذلك لمواجهة خطر التهديد المتصاعد لانتشار الفيروسات من خلال الإنترنت. في البداية سنلقي بعض الضوء على هذا التهديد، ثم سنُلخِّص طريقة IBM.

بشكلٍ تقليدي، كانت تهديدات الفيروسات تتسم بالانتشار البطيء نسبياً للفيروسات الجديدة والتغيرات الجديدة. فعادةً ما كان يتم تحديث البرامج المضادة للفيروسات بشكل شهري، وكان هذا كافياً للسيطرة على المشكلة. أيضاً بشكل تقليدي، أدّت الإنترنت دوراً صغيراً نسبياً في انتشار الفيروسات. لكن كما بيَّن [CHES97] هناك اتجاهان رئيسان في تقنية الإنترنت كان لهما تأثيرٌ متزايدٌ على معدل انتشار الفيروسات في السنوات الأخيرة:

- أنظمة البريد المدمجة: جعلت أنظمة البريد مثل Notes من Lotus وOutlook من مايكروسوفت إرسال أي شيء لأي أحد، وكذلك معالجة الكائنات التي تُستلم، أمراً بسيطاً للغاية.
- أنظمة البرامج النقَّالة: سمحت التقنيات الحديثة مثل Java وActiveX للبرامج بالانتقال تلقائياً من نظام إلى آخر.

لمواجهة التهديدات الناجمة عن تلك الإمكانات المبنية على الإنترنت، طورت شركة IBM نموذجاً لنظام المناعة الرقمي. يمثل هذا النظام امتداداً لاستخدام المحاكاة وقد نوقش في الجزء الفرعي السابق، ويوفّر نظاماً متعدد الأغراض للمحاكاة وكشف الفيروسات. يهدف النظام إلى تسريع الاستجابة بحيث يتم إخماد الفيروسات بمجرد ظهورها تقريباً. وعندما يدخل فيروس جديد إلى مؤسسةٍ ما، يقوم نظام المناعة آلياً بأسره، وتحليّله، بالإضافة إلى كشفه والحماية منه، ثم إزالته، وتمرير المعلومات الخاصة به إلى الأنظمة التي تستخدم برامج IBM المضادة للفيروسات لاكتشافه قبل أن تتاح له فرصة إتلاف مكان آخر.

يوضح الشكل 10-4 الخطوات المعتادة لطريقة عمل نظام المناعة الرقمي:

1. يستخدم برنامج المراقبة على كل جهاز حاسب شخصي تشكيلة من القواعد الاجتهادية للكشف عن وجود الفيروس بناءً على سلوك النظام، أو التغييرات المريبة في البرامج، أو التوقيع المميِّز لسُلالة الفيروس. يُرسِل برنامج المراقبة نسخة من أي برنامج يعتقد أنه مصاب إلى جهاز حاسب إداري بالمؤسسة.
2. يُشفِّر جهاز الحاسب الإداري العيّنة ويرسلها إلى جهاز حاسب مركزي لتحليل الفيروسات.
3. ينشئ ذلك الجهاز بيئةً يمكن فيها تشغيل البرنامج المصاب للتحليل بأمان. تتضمّن الأساليب المستخدمة لهذا الغرض المحاكاة، أو إنشاء

بيئة محميّة لتنفيذ البرنامج المشكوك فيه ومراقبته. بعد ذلك يُنتِج جهاز تحليل الفيروسات وصفة للتعرف على الفيروس وإزالته.

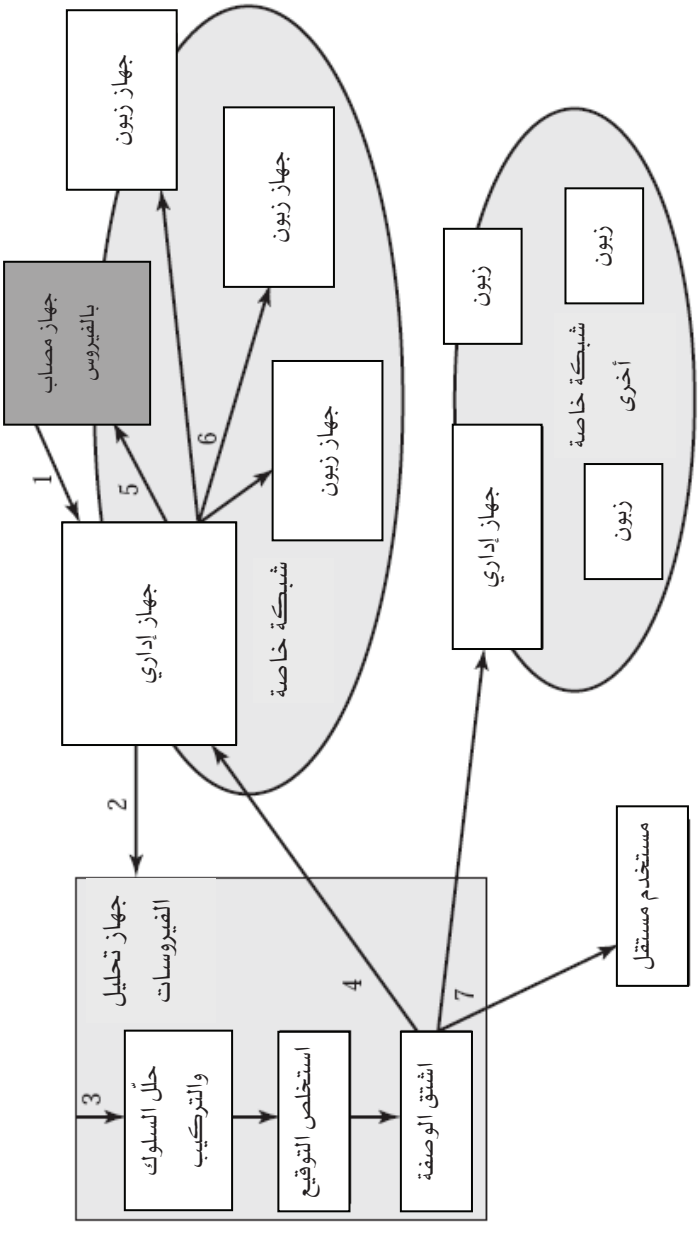
4. تُرسَل الوصفة الناتجة إلى جهاز الحاسب الإداري.
5. يُرسِل جهاز الحاسب الإداري الوصفة إلى الزبون المصاب.
6. تُرسَل الوصفة أيضاً إلى الزبائن الأخرى في المؤسسة.
7. يتلقّى المشتركون حول العالم بصفة منتظمة تحديثات للبرامج المضادة للفيروسات للحماية من الفيروسات الجديدة.

يعتمد نجاح نظام المناعة الرقمي على قدرة الجهاز الذي يقوم بتحليل الفيروسات على اكتشاف السلالات الجديدة والمبتكرة من الفيروسات. بتحليل ومراقبة الفيروسات التي تقبع في البرِّية، سيتسنى تحديث برامج المناعة الرقمية باستمرار لمواكبة التهديدات الجديدة.

10-2-3 برامج منع السلوك المشبوه (Behavior-Blocking Software)

على خلاف برامج المسح التي تعتمد على القواعد الاجتهادية أو على البصمة (توقيع الفيروس)، تُدمج برامج منع السلوك المشبوه مع نظام تشغيل الحاسب المضيف لتراقب سلوك البرامج بشكل آني لاكتشاف الأفعال المؤذية. عندئذ تقوم تلك البرامج بمنع الأفعال المؤذية المحتملة قبل أن تتمكّن من التأثير على النظام. يمكن أن يتضمّن السلوك المراقَب ما يأتي:

- محاولات فتح الملفات أو عرضها أو حذفها أو تعديلها.
- محاولات تهيئة (formatting) مشغل الأقراص وغير ذلك من عمليات الأقراص الأخرى غير القابلة للاسترداد.
- تعديلات لمنطق الملفات التنفيذية أو الملفات التي تتضمن ماكرو.
- تعديل إعدادات النظام الحرجة، كإعدادت بدء التشغيل.



الشكل 10-4: نظام مناعة رقمي.

- طلب إرسال محتوى قابل للتنفيذ بالبريد الإلكتروني أو عبر المراسلة الإلكترونية الفورية.
- إنشاء اتصال عبر الشبكة.

إذا اكتشف مانع السلوك المشبوه (behavior blocker) أن برنامجاً يشرع فيما يمكن أن يؤول إلى سلوكٍ مؤذٍ أثناء تشغيله، يمكنه منع هذا السلوك على الفور وإنهاء البرنامج المشبوه. ويُحقق ذلك ميزة أساسية على طرق الكشف المعروفة المضادة للفيروسات والمبنية على أساس البصمات أو القواعد الاجتهادية. فبينما يوجد حرفياً تريليونات من الطرق المختلفة لتشويش وإعادة ترتيب تعليمات الفيروس أو الدودة، كثيرٌ منها لن يتسنى اكتشافه من قِبَل ماسحات البصمات أو القواعد الاجتهادية، فإن أي كود مؤذٍ لابد في النهاية وأن يرسل طلباً واضح المعالم إلى نظام التشغيل. بافتراض أن مانع السلوك يستطيع اعترض كل مثل تلك الطلبات، فسيمكنه أن يكتشف ويمنع الأعمال المؤذية بغض النظر عن كيفية التشويه الذي تم إدخاله على منطق البرنامج للتمويه.

واضحٌ أنَّ القدرة على مراقبة البرامج أثناء تشغيلها في الوقت الحقيقي يحقق ميزةً كبيرةً لمانع السلوك المشبوه؛ ومع ذلك فإن لها مثالب أيضاً. نظراً لأنه ينبغي تشغيل الكود المؤذي على الجهاز المستهدف قبل أن يمكن تمييز كل جوانب سلوكه، فقد يتسبب في إلحاق كثير من الضرر بالنظام قبل اكتشافه ومنعه. على سبيل المثال، قد يخلط فيروس جديد عدداً من ملفات قد تبدو غير مهمة على القرص الصلب قبل إصابة أي ملف وقبل منعه. ورغم أن العدوى الفعلية قد أمكن منعها، فقد يعجز المستخدم عن تحديد مواقع ملفاته، مما يسبب فقداً في الإنتاجية أو يؤدي إلى عواقب أسوأ من ذلك.

3-10 هجمات حجب الخدمة الموزَّعة

تمثل هجمات حجب الخدمة الموزَّعة (DDoS) تهديداً أمنياً بالغ الخطورة للشركات، ويبدو أن خطر هذا التهديد في تزايد [VIJA02]. وفي إحدى الدراسات التي غطت فترة ثلاثة أسابيع في عام 2001، رصد الباحثون أكثر من 12,000 هجمة ضد أكثر من 5000 هدف محدّد. وتراوحت الأهداف من الشركات المشهورة للتجارة عبر الإنترنت كالأمازون والهوتميل إلى موفري خدمة الإنترنت (ISP) الأجانب الصغار ووصلات الاتصال عبر الهاتف [MOOR01]. وتجعل هجمات DDoS أنظمة الحاسب صعبة الوصول وذلك بغمر الخادمات، أو الشبكات، أو حتى الأنظمة الطرفية بحركة مرور عديمة الفائدة بحيث لا يستطيع المستخدمون الشرعيون الوصول إلى تلك الموارد. في هجوم DDoS نمطي يُطلب من عدد كبير من المضيفات المصابة إرسال رزمٍ عديمة الفائدة. غير أنه في السنوات الأخيرة أصبحت أساليب الهجوم وأدواته أكثر تطوراً وفعاليةً، وصار تتبّع أثر المهاجمين الحقيقيين أكثر صعوبةً، بينما بقت تقنيات الدفاع عاجزةً عن مقاومة الهجمات واسعة النطاق [CHAN02].

هجوم حجب الخدمة ((denial of service (DoS) هو محاولة لمنع مستخدمين شرعيين لخدمةٍ ما من استخدام تلك الخدمة. وعندما يَصدُر هذا الهجوم من مضيف واحد أو عقدة واحدة على الشبكة، يطلق عليه ببساطة هجوم DoS. ويُعدُّ هجوم حجب الخدمة الموزّع (DDoS) أكثر خطورة، حيث يكون المهاجم قادراً على تجنيد عددٍ من المضيفات في كافة أنحاء الإنترنت بشكلٍ آني ومنسّق للهجوم على الهدف. وسنركّز في هذا الجزء على هجمات DDoS. سننظر في البداية إلى طبيعة تلك الهجمات وأنواعها. ثم نتناول الوسائل التي تُمكِّن المهاجم من تجنيد مضيفات على الشبكة لإطلاق الهجوم. وأخيراً نناقش الإجراءات المضادة.

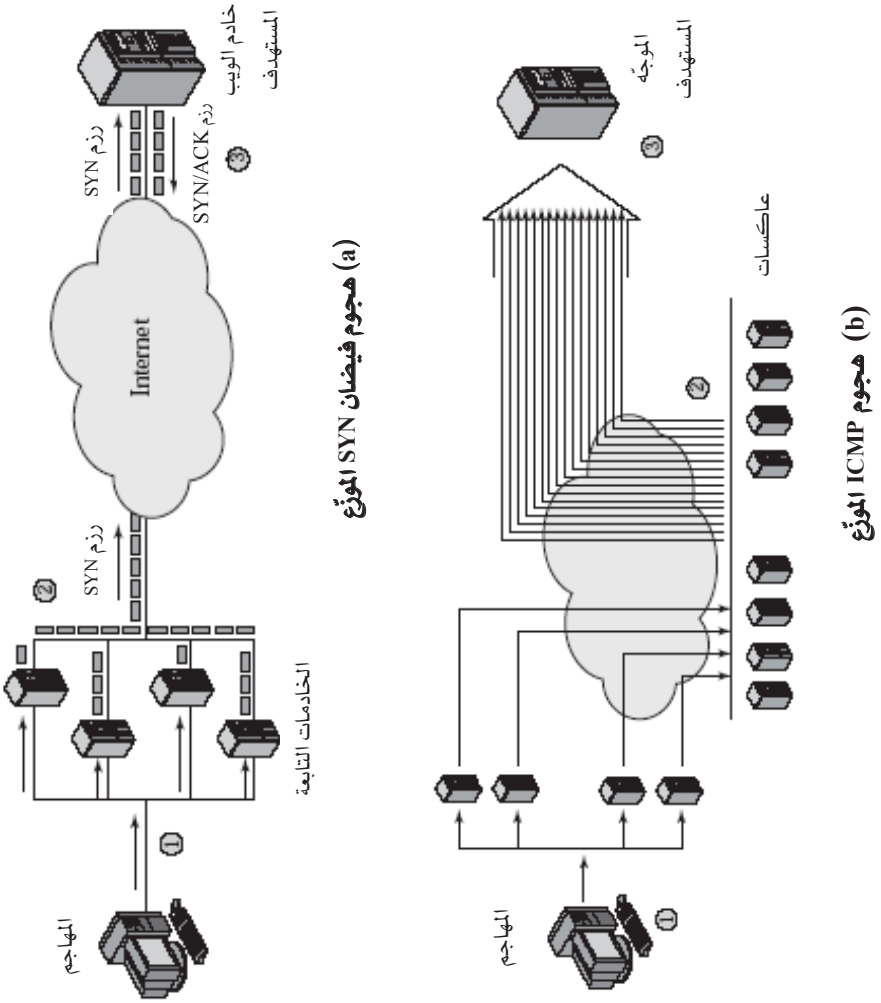
10-3-1 وصف هجوم DDoS

يحاول هجوم DDoS استهلاك موارد الهدف حتى لا يستطيع توفير الخدمة. وإحدى الطرق لتصنيف هجمات DDoS هي تصنيفها من حيث نوع المورد الذي يُستهلك. بشكل عام، يكون المورد الذي يُستهلك إما مورداً داخلياً على نظام المضيف المُستهدف، أو سعة نقل بيانات على الشبكة المحليّة الموصّل عليها ذلك المضيف. من الأمثلة البسيطة لهجوم مورد داخلي هجوم فيضان SYN.

يوضح الشكل 10-5 (a) الخطوات التي يتضمنها ذلك الهجوم:

1. أولاً يسيطر المهاجم على عدة مضيفات على الإنترنت، ويأمرها بالاتصال بخادم ويب الهدف.
2. تبدأ تلك المضيفات التابعة (slave hosts) بإرسال رزم SYN – للتزامن والتهيئة في بروتوكول TCP/IP – إلى الهدف بعنوان IP مزوّر للعودة.
3. تمثّل كل رزمة SYN طلباً لإنشاء توصيلة TCP. لكل رزمة من تلك الرزم، يردّ خادم الويب للمضيف المستهدف برزمة SYN/ACK (تزامن وإشعار استلام)، محاولاً تأسيس توصيلة TCP مع كيان TCP بعنوان IP المزوّر.

يحتفظ خادم الويب بهيكل بيانات لكل طلب SYN ينتظر الردّ، ومن ثم يصبح متورِّطاً أكثر كلما ازداد فيضان مرور بيانات. وتكون النتيجة عدم تلبية الطلبات لتوصيلات شرعية بينما يكون المضيف الضحيَّة ينتظر إكمال عددٍ كبيرٍ من التوصيلات المزيفة "نصف المفتوحة".



الشكل 10-5: أمثلة لهجمات DDoS البسيطة.

كثيراً ما يُستخدم هيكل بيانات حالة بروتوكول TCP كمورد داخلي مستهدف للهجوم، ولكنه ليس الوحيد. يعطي [CERT01] الأمثلة الآتية:

1. في كثير من الأنظمة يتوفر عددٌ محدود من هياكل البيانات لتخزين معلومات العملية (process) (كمُعرِّفات العملية، ومُدخلات جداول العملية، إلخ). قد يتمكن مهاجم من استهلاك هياكل البيانات تلك بكتابة برنامج بسيط لا يفعل شيئاً سوى إنشاء نسخة من نفسه مراراً وتكرارا.
2. قد يحاول المهاجم أيضاً استهلاك سعة القرص بطرق أخرى، مثل:
   - إنشاء عدد كبير جداً من رسائل البريد الإلكتروني.
   - تعمُد إنشاء أخطاء يقوم النظام بتسجيلها.
   - وضع ملفات في مناطق مجهولة بنظام نقل الملفات (ftp) أو مناطق مشتركة على الشبكة.

يوضح الشكل 10-5 (b) مثالاً لهجوم يستهلك موارد نقل البيانات، ويشمل الخطوات الآتية:

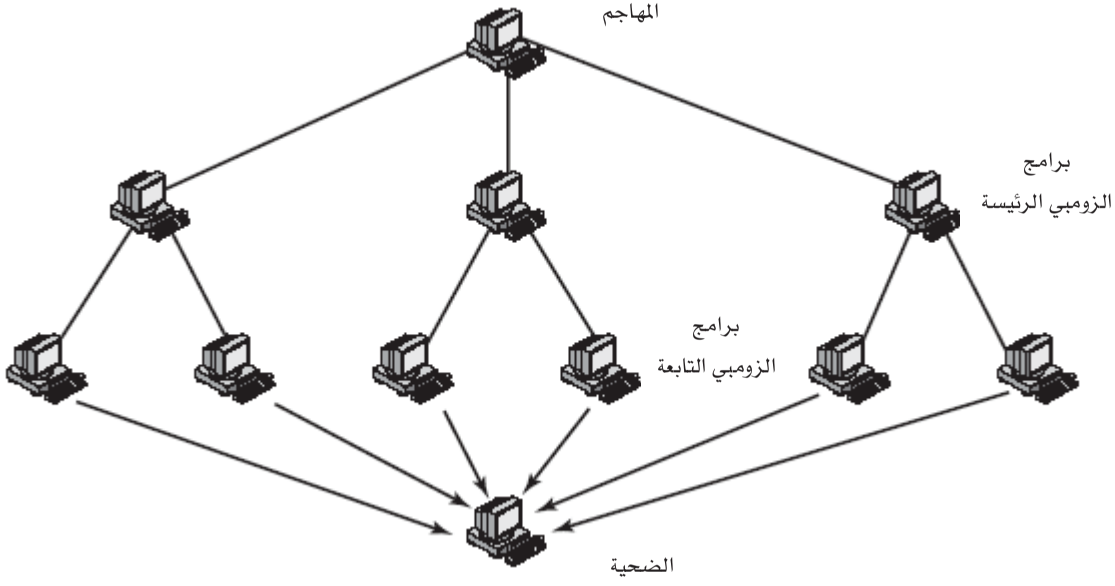
1. يسيطر المهاجم على عدة مضيفات على الإنترنت، ويأمرها بإرسال رزم صدى ICMP[1] بعنوان IP للهدف (كعنوان مزيف (spoofed address) للمصدر) إلى مجموعة من المضيفات التي تعمل بمثابة عاكسات (reflectors)، كما سنصف بعد ذلك.
2. تستلم العقد في موقع الإرجاع (bounce site) عدة طلبات مزيفة وترد بإرسال رزم رد للصدى إلى موقع الهدف.

[1] بروتوكول رسائل التحكم للإنترنت ((Internet Control Message Protocol (ICMP) هو بروتوكول في طبقة الشبكة (مستوى بروتوكول IP) لتبادل رزم التحكم بين موجّه ومضيف أو بين المضيفات. تتطلب رزمة الصدى (ECHO) من المستقبل أن يرد برزمة الاستجابة للصدى (echo reply) مما يفيد في فحص الاتصال بين كيانات مختلفة.

3. يصاب موجّه الهدف بفيضان الرزم من موقع الإرجاع، ومن ثم لا تبقى سعة كافية لنقل حركة البيانات الشرعية.

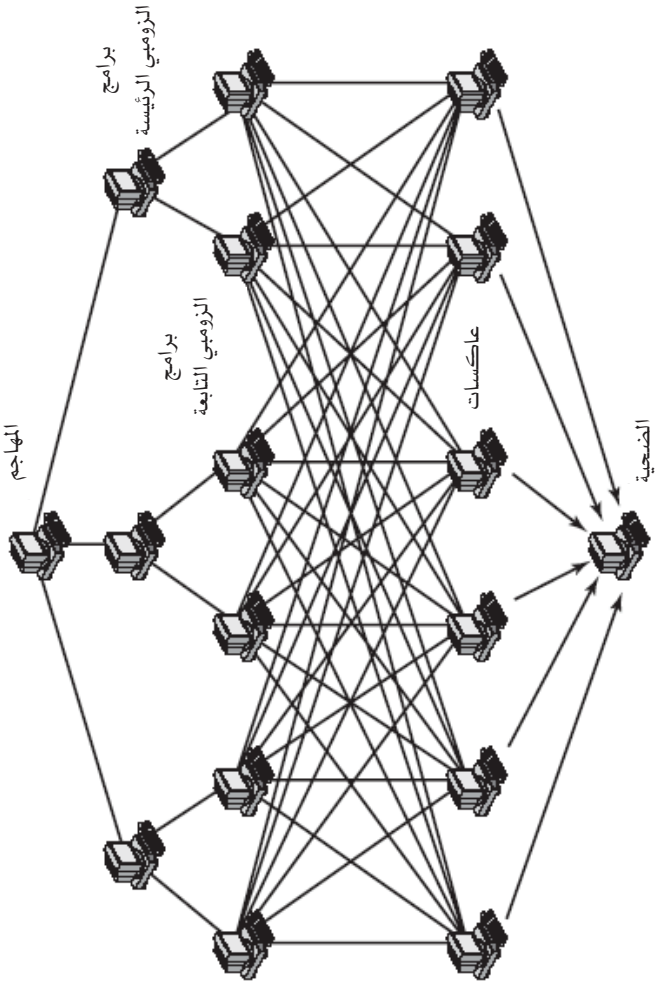
من طريق التصنيف الأخرى تصنيف هجمات DDoS كهجمات مباشرة (direct) أو هجمات انعكاسية (reflector). في هجوم DDoS المباشر (الشكل 10-6 (a))، يستطيع المهاجم زراعة برامج الزومبي على عدد من المواقع الموزَّعة في كافة أنحاء الإنترنت. في أغلب الأحيان، يتضمّن هجوم DDoS مستويين من مضيفات الزومبي: الزومبي الرئيس والزومبي التابع. وتُصاب المضيفات في كلا المستويين بالبرمجيات الخبيثة. ويوجِّه المهاجم مضيفات الزومبي الرئيسة وينسق فيما بينها، والتي تقوم بدورها بالإثارة والتنسيق فيما يتعلق بمضيفات الزومبي التابعة. إن استخدام مستويين من الزومبي يزيد من صعوبة تتبّع الهجوم للوصول إلى مصدره الأصلي، ويوفر بيئة مرنة لشبكة المهاجمين.

يضيف هجوم DDoS الانعكاسي طبقةً أخرى من المضيفات (الشكل 10-6 (b)). في هذا النوع من الهجوم، تنشئ مضيفات الزومبي التابعة رزماً تتطلب رداً يتضمن عنوان IP للهدف كعنوان IP للمصدر في ترويسة العنوان بالرزمة. وتُرسَل تلك الرزم إلى الأجهزة غير المصابة والمعروفة بالعاكسات. ترد الأجهزة غير المصابة برزم موجّهة نحو جهاز الهدف. يمكن أن يتضمّن هجوم DDoS الانعكاسي بسهولة مضيفات أكثر وحركة مرور بيانات أكبر من هجوم DDoS المباشر، ومن ثَمَّ يكون أكثر ضرراً. بل إن تتبّع الهجوم إلى مصدره الأصلي أو ترشيح رزم الهجوم يكون أكثر صعوبة؛ لأن الهجوم يأتي من أجهزة غير مصابة ومنتشرة على مناطق كبيرة.



(a) هجوم DDoS مباشر

الشكل 10-6 أنواع هجمات DDoS التي تعتمد على الفيضان.



(b) هجوم DDoS انعكاسي

تابع الشكل 10-6 أنواع هجمات DDoS التي تعتمد على الفيضان.

10-3-2 إنشاء شبكة الهجوم

في الخطوة الأولى من هجوم DDoS يتمكن المهاجم من إصابة عددٍ من الأجهزة ببرامج الزومبي التي ستُستخدم في النهاية لتنفيذ الهجوم. تضم المكونات الأساسية اللازمة للهجوم في هذه المرحلة ما يأتي:

1. البرامج التي يمكنها تنفيذ هجوم DDoS. يجب أن تصلح تلك البرامج لتشغيلها على عدد كبير من الأجهزة، وأن تكون قادرة على إخفاء وجودها، وأن تكون قادرة على الاتصال مع المهاجم أو يكون لها نوع من المثيرات الآلية الموقوتة، وأن تكون قادرة على شن الهجوم المقصود نحو الهدف.
2. نقطة ضعف في عددٍ كبيرٍ من الأنظمة. يجب أن يتعرف المهاجم على نقطة ضعفٍ أمنيةٍ فَشَل كثيرٌ من مديري الأنظمة والمستخدمين في ترقيعها ويستطيع المهاجم من خلالها تثبيت برامج الزومبي على تلك الأنظمة.
3. استراتيجية لتحديد أماكن الأجهزة التي تعاني من نقاط الضعف، وتُعرف هذه العملية بالمسح.

في عملية المسح، يبحث المهاجم أولاً عن عددٍ من الأنظمة التي تعاني من نقاط الضعف ويصيبها. وعادةً ما تُكرر برامج الزومبي التي تُثبَّت على الأنظمة المصابة بتكرار نفس عملية المسح إلى أن يتم تكوين شبكة كبيرة من الأنظمة الموزّعة المصابة. استعرَض [MIRK04] الأنواع الآتية من استراتيجيات المسح:

- المسح العشوائي: يتقصّى كل مضيف مصاب عناوين IP في فضاء العناوين بطريقة عشوائية باستخدام بذرة (seed) مختلفة. تُنتج هذه

الطريقة كميةً ضخمةً من حركة مرور بيانات الإنترنت، مما قد يسبب عرقلة عامة حتى قبل شن الهجوم الفعلي.

- المسح باستخدام قائمة الإصابة (hit-list): يجمع المهاجم أولاً قائمة طويلة من الأنظمة الضعيفة المحتملة. ويمكن أن تكون هذه عملية بطيئة وتتم على فترة طويلة لتفادي اكتشاف أن هجوماً يجري تنفيذه. وبمجرد تكوين القائمة، يبدأ المهاجم بإصابة الأنظمة الموجودة بالقائمة. ويُزوّد كل نظام مصاب بجزء من القائمة ليقوم بمسحها. وتؤدّي هذه الاستراتيجية إلى فترة مسح قصيرة جداً، مما قد يجعل من الصعب اكتشاف أن العدوى جارٍ انتشارها.
- المسح بشكل طوبولوجي: تستخدم هذه الطريقة معلومات موجودة على النظام الضحيّة لمعرفة المزيد من المضيفات لمسحها.
- مسح الشبكة الفرعية المحليّة: إذا أمكن إصابة مضيف خلف برنامج الحماية (الجدار الناري) (firewall)، يقوم ذلك المضيف بعد ذلك بالبحث عن أهداف ضمن شبكته المحليّة. يستخدم المضيف هيكل عناوين الشبكة الفرعية لإيجاد مضيفات أخرى محميَّة ببرنامج الجدار الناري (ضد الطرق الأخرى).

10-3-3 الإجراءات المضادة لهجوم DDoS

بشكلٍ عام هناك ثلاثة من خطوط الدفاع ضد هجمات DDoS [CHAN02]:

- منع الهجوم وإحباطه (قبل الهجوم): تُمكّن هذه الآليات النظام الضحيّة من تحمّل محاولات الهجوم بدون حجب الخدمة عن الزبائن الشرعيين. وتتضمّن الأساليب المستخدمة فرض سياسات لاستهلاك الموارد وتوفير

موارد احتياطية متوفرة عند الطّلب. بالإضافة لذلك، تُعدِّل تلك الآليات الأنظمة والبروتوكولات على الإنترنت لتقليل احتمالات هجمات DDoS.

- اكتشاف الهجوم والترشيح (أثناء الهجوم): تحاول هذه الآليات اكتشاف الهجوم بمجرد أن يبدأ وتستجيب لذلك فوراً، مما يقلّل من تأثير الهجوم على الهدف. ويتضمّن الكشف عن الهجوم البحث عن أنماط السلوك المريبة. ويتضمّن الردّ ترشيح رزم يُحتمل أن تكون جزءاً من الهجوم.
- تتبُّع مصدر الهجوم والتعرف عليه (أثناء وبعد الهجوم): تلك محاولة لمعرفة مصدر الهجوم كخطوة أولى لمنع الهجمات في المستقبل. ومع ذلك، لا تؤدي هذه الطريقة عادةً إلى نتائج سريعةً بما فيه الكفاية لتسكين هجوم مستمر، وقد لا تكون لها نتائج تُذكر.

يكمن التحدي في التعامل مع هجمات DDoS في الطرق الكثيرة التي يمكن أن تحدث بها. ولذا ينبغي أن تتطوَّر إجراءات المواجهة باستمرار لتواكب التقدم في أساليب الهجوم.

4-10 توصيات للمطالعة

[CASS01] Cass, S. "Anatomy of Malice." *IEEE Spectrum*, November 2001.

[CHAN02] Chang, R. "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial." *IEEE Communications Magazine*, October 2002.

[FORR97] Forrest, S.; Hofmeyr, S.; and Somayaji, A. "Computer Immunology." *Communications of the ACM*, October 1997.

[HARL01] Harley, D.; Slade, R.; and Gattiker, U. *Viruses Revealed*. New York: Osborne/McGraw-Hill, 2001.

[KEPH97] Kephart, J.; Sorkin, G.; Chess, D.; and White, S. "Fighting Computer Viruses." *Scientific American*, November 1997.

[MEIN01] Meinel, C. "Code Red for the Web." *Scientific American*, October 2001.

[MIRK04] Mirkovic, J., and Relher, P. "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms." *ACM SIGCOMM Computer Communications Review*, April 2004.

[NACH97] Nachenberg, C. "Computer Virus-Antivirus Coevolution." *Communications of the ACM*, January 1997.

[PATR04] Patrikakis, C.; Masikos, M.; and Zouraraki, O. "Distributed Denial of Service Attacks." *The Internet Protocol Journal*, December 2004.

[SZOR05] Szor, P., *The Art of Computer Virus Research and Defense*.Reading, MA: Addison-Wesley, 2005.

5-10 مصادر للمعلومات على الويب

- AntiVirus Online: موقع لشركة IBM يتضمن معلومات عن الفيروسات.
- Vmyths: موقع مخصص لعرض خدع الفيروسات (virus hoaxes) وتبديد المفاهيم الخاطئة عن الفيروسات الحقيقية.
- DDoS Attacks/Tools: قائمة واسعة من الوصلات التشعبية (عناوين الويب) والوثائق.

6-10 مصطلحات رئيسة

auto-rooter	برنامج للوصول التلقائي للجذر
backdoor	مدخل خلفي
digital immune system	نظام مناعة رقمي
direct DDoS attack	هجوم حجب الخدمة الموزَّع المباشر
distributed denial of service (DDoS)	حجب الخدمة الموزَّع
downloaders	برامج التنزيل
e-mail virus	فيروس البريد الإلكتروني
exploits	استغلالات (نقاط ضعف)
flooder	مُرسِل الفيضان
keyloggers	مسجلات ضربات المفاتيح
kit	علبة أدوات (لتوليد فيروسات)
logic bomb	قنبلة منطقية
macro virus	فيروس الماكرو
malicious software (malware)	برمجيات مؤذية
polymorphic virus	فيروس متعدد الأشكال
reflector DDoS attack	هجوم DDoS الانعكاسي
rootkit	الجذور الخفية (أو أدوات الجذر)
spammer program	برنامج إرسال رسائل الدعاية
stealth virus	فيروس مُتَخفٍّ
trapdoor	باب المصيدة (المدخل الخلفي)
Trojan horse	حصان طروادة
virus	فيروس
worm	دودة
zombie	برنامج زومبي

7-10 أسئلة للمراجعة ومسائل

1-7-10 أسئلة للمراجعة

1-10 ما دور الضغط (compression) في طريقة عمل الفيروس؟

2-10 ما دور التشفير في طريقة عمل الفيروس؟

3-10 ما المراحل المعتادة لعمل الفيروس والدودة؟

4-10 بشكلٍ عام كيف تنتشر الديدان؟

5-10 ما المقصود بنظام المناعة الرقمي؟

6-10 كيف تعمل برامج منع السلوك المشبوه؟

7-10 ماذا تعرف عن هجوم حجب الخدمة الموزَّع (DDoS)؟

2-7-10 مسائل

1-10 يوجد خلل في برنامج الفيروس الموجود بالشكل 10-1؛ ما هو؟

2-10 السؤال الذي يطرح نفسه هو ما إذا كان من الممكن تطوير برنامج يستطيع تحليل قطعة من البرمجيات ويحدد ما إذا كانت تمثل فيروساً أم لا؟ اعتبر أن لدينا برنامج D يُفترض أنه قادر على ذلك. بمعنى أنه لأي برنامج P، إذا شغّلنا D(P) فإن النتيجة تكون صح (true) في حالة ما إذا كان P فيروساً أو خطأ (false) إذا لم يكن P فيروساً. الآن خذ في الاعتبار البرنامج الآتي:

```
program CV :=
{ . . .
main-program :=
{
if D(CV) then
      goto next;
else
      infect-executable;
}
next:
}
```

في هذا البرنامج تُعدُّ infect-executable وحدة برمجية لمسح الذاكرة بحثاً عن برامج تنفيذية ونسخ نفسها في تلك البرامج. حدد ما إذا كان D يمكنه أن يقرر بدقة ما إذا كان CV عبارة عن فيروس أم لا.