VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

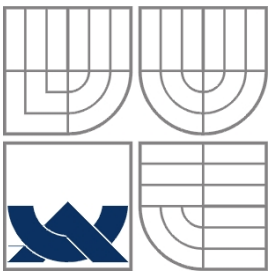
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KRYPTOVIROLOGIE A BUDOUCNOST MALWARE

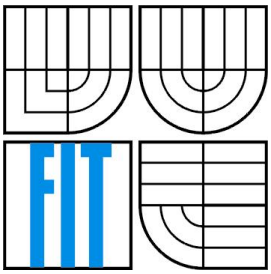
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE JOSEF PRCHAL
AUTHOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KRYPTOVIROLOGIE A BUDOUCNOST MALWARE

CRYPTOVIROLOGY AND FUTURE OF MALWARE

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

JOSEF PRCHAL

VEDOUCÍ PRÁCE
SUPERVISOR

DANIEL CVRČEK

BRNO 2007

Abstrakt

Malware je spojený s informační technikou. Oboje se ovlivňuje navzájem. Cílem této práce je přiblížit různé typy těchto programů, stručně popsat historii a vývoj. Nastínit hlavní trendy v této oblasti a pokusit se předpovědět, kam bude vývoj směřovat.

Klíčová slova

A pack with the Devil, bezpečnost, bot, botnet, červ, distribuovaný výpočet, hoax, Hybris, integrace kódu, IRC, kryptovirologie, malware, metamorfismus, Morrisův červ, neuronové sítě, OneHalf, openssl, P2P, peer-to-peer, polymorfismus, Slapper, sociální inženýrství, válka červů, Vecna, vir, virus, Zmist, Zombie

Abstract

Malware is connected to information technology. They influence each other. The aim of this thesis is to describe various types of this software and give a brief account of its history and development. It also discusses main trends of this area and tries to foretell the future development.

Keywords

A pack with the Devil, bot, botnet, code integration, cryptovirology, distributed computing, hoax, Hybris, IRC, malware, metamorphic, Morrisův červ, neural network, OneHalf, openssl, P2P, peer-to-peer, polymorphic, security, Slapper, social engineering, Vecna, virus, war of the worms, worm, Zmist, Zombie

Citace

Josef Prchal: Kryptovirologie a budoucnost malware, bakalářská práce, Brno, FIT VUT v Brně, 2007

Kryptovirologie a budoucnost malware

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jméno Příjmení
Datum

Poděkování

Děkuji vedoucímu své práce, panu Danielu Cvrčkovi, za cenné rady, připomínky a materiály, které mi při psaní práce poskytl. Další dík patří mé rodině za podporu.

Obsah

1 Úvod

Jak z názvu vyplývá, tato práce se bude týkat především malware. V dnešní době se informační technologie (IT) a s ní související obory rozvíjí velmi rychlým tempem. Počítače jsou všude kolem nás a stávají se nedílnou součástí našeho života. Bez internetu by si už většina lidí nedokázala život ani představit. Nejen, že ho potřebují pro zábavu, k vyhledávání informací, ke komunikaci s okolím a v zaměstnání, ale zároveň na něm závisí i většina společností. A pokud na něm nezávisí přímo, tak dozajista potřebují ke své existenci jinou společnost, jež se bez internetu neobejde.

Současně se rozvíjí i nový druh kriminality. Stejně jako mince má i tato oblast rub a líc. Temnou stranou IT jsou škodlivé programy. Proto jsem si vybral za téma své bakalářské práce kryptovirologii a budoucnost malware. Rád bych popsal a zhodnotil odkud a kam malware kráčí.

Na začátku mé práce naznačím jedno z mnoha možných rozdělení malware. Já použiji klasifikaci dle způsobu šíření. Je ale zřejmé, že mnoho malware lze zařadit do více různých kategorií.

V další kapitole představím časovou osu, na které uvedu známější škodlivé programy s roky, kdy se poprvé objevily. Popíši také některé metody, které malware používá ke své ochraně proti detekci antiviry.

Poté bych rád představil čtyři malware, které mě při psaní práce nejvíce zaujaly. Jsou to: viry OneHalf a Zmist, červi Hybris a Slapper. Jejich jedinečné vlastnosti popíši v této práci podrobněji.

Následující kapitolu věnuji sociálnímu inženýrství. Uvedu zde lidské rysy, které malware pomocí sociálního inženýrství využívá. Na příkladě „ďábelského“ malware, jenž využívá špatné vlastnosti lidí a dokonce nabízí spolupráci, názorně ukáži, jak mu takovýto pakt s uživatelem umožňuje alternativní cesty šíření a zaručuje mu vyšší bezpečnost před odstraněním.

V poslední kapitole pak pomocí současných trendů naznačím směr, kterým by se mohl malware ubírat. Přes boty vytvářející vlastní P2P sítě, po boj o cíle útoků a využívání chyb v samotných škodlivých programech. Malou část věnuji i distribuovaným výpočtům, použitelných při útocích na samotné šifry. Nesmím zapomenout i na kryptovirologii. Kapitolu pak ukončím výčtem rysů nutných k přežití malware v dnešní době.

2 Základní definice a rozdělení malware

Malware je škodlivý počítačový program, obvykle určený ke vniknutí nebo poškození počítačového systému. Výraz *malware* vznikl složením anglických slov „malicious“ (zákeřný) a „software“ (programové vybavení) a popisuje spíše záměr autora takového programu než jeho specifické vlastnosti. Je zřejmé, že každá klasifikace malware narazí na problémy, protože se jeho různé třídy překrývají a často představují příbuznou podtřídu nějaké jiné.

Níže uvedené dělení malware je jedno z mnoha možných [1]:

1. **Viry (Viruses)**

   „Počítačový virus je program, který rekurzivně a explicitně kopíruje potenciálně se vyvíjející verzi sebe sama.“ [1]

   Virus (virus = latinsky jed) je struktura nacházející se na hranici mezi živým a neživým. Ty nejprimitivnější viry obsahují pouze svoji genetickou informaci ve formě DNA nebo RNA, které jsou uloženy v kapsidě, a několik málo proteinů tvořících virový obal. Ty složitější mohou navíc obsahovat 1-2 obalové membrány pocházející z napadené buňky a enzymy, které jim mají usnadnit invazi do buňky a expresi své DNA či RNA. Viry nejsou schopny samostatné existence bez hostitelské buňky, tedy přesněji, nejsou schopny se bez hostitelské buňky reprodukovat. [2]

   Z této definice biologického viru je zřejmé, proč byl tento termín použit pro pojmenování počítačových virů. Počítačový virus je sled instrukcí/funkcí, díky kterým se replikuje, bez hostitele to však není možné. Mezi hostitele patří například spustitelný soubor, systémová oblast disku nebo soubory, které se přímo nevykonávají, nýbrž jsou interpretovány jiným programem – skripty, dokumenty Microsoft Word. Složitější viry obsahují více metod šíření a sebe-replikace, případně i různé formy obrany a útoku, které jsou využívány pro své přežití.

2. **Počítačoví červi (Worms)**

   Počítačovými červy označujeme síťové viry. Šíří se dvěma cestami. Buď se samy, bez zásahu ze strany uživatele, dokáží spustit na vzdáleném počítači využitím bezpečnostní díry v jeho systému nebo spuštěném programu (z infikovaného počítače se šíří náhodně či podle nějakého algoritmu na další počítače, což může vést až k zahlcení sítě). Nebo se šíří pomocí

e-mailů, v tomto případě červi pomoc ke spuštění uživatele vyžadují. Nabízejí zajímavý obsah, neznalý uživatel jim uvěří a spustí je. Protože to jsou samostatné programy, nepotřebují pro běh hostitele.

3. **Logické bomby (Logic bombs)**

   Logická bomba je kus kódu vloženého do programu, který spustí zákeřnou funkci za předpokladu, že jsou splněny určité podmínky. Řadí se sem programy, které se po určitém počtu spuštění samy smažou z disku, jako ochrana proti kopírování. Dále sem náleží programy, jež po daném čase zobrazí reklamu a podobně. U větších projektů to mohou být také oblíbená velikonoční vajíčka (easter eggs). Například v balíčku Microsoft Office je uveden seznam členů týmu, který pracoval na vývoji produktu.

4. **Trojští koně (Trojan Horses)**

   Jak už název vypovídá, jde o program vydávající se za užitečný, avšak ve skutečnosti je škodlivý. Trojský kůň může být samostatný program nevyžadující hostitele. Z důvodu lehčího šíření se ale často používá obyčejný program obohacený o trojského koně. Pokud je takový program spuštěn, uživatel vidí pouze onen obyčejný program a trojský kůň běží na pozadí. Smyslem těchto programů je často vytvoření zadních vrátek (backdoors/trapdoors), tedy umožnění útočníkovi přistupovat do napadeného systému. Další možností je třeba odchytávání hesel, kdy trojský kůň buď vyhledává hesla přímo na disku nebo zaznamenává stisky kláves na klávesnici a pak je odesílá útočníkovi.

5. **Zárodky (Germs)**

   Zárodky představují první generaci virů, které ještě nejsou plně funkční, obvykle právě zkompilované, existující bez hostitele, nezašifrované a s čitelným kódem.

6. **Exploity (Exploits)**

   Do této kategorie spadají programy, které dokáží vniknout do systému pomocí konkrétní zranitelnosti. Snaží se, aby byly spuštěny nebo aby získaly vysoce privilegovaný přístup. Závisí na útočníkovi, zda exploit použije k šíření nebezpečného malware.

7. **Stahovače (Downloaders)**

   Tento nepřímo škodlivý kód, sám o sobě nezávadný, je určen ke stažení a spuštění jiného malware.

8. **Dialery (Dialers)**

   Škodí tím, že změní u vytáčeného připojení (dial-up) daný tarif za jiný, dražší. Nemusí se nutně vyskytovat jako samostatný program, může být jako skript součástí html stránky.

9. **Droppery (Droppers)**

   Původně to byly instalátory první generace virů. Staraly se o zavedení viru do boot sektoru disku nebo prvotně zašifrovaly vir a připojily ho k hostiteli. Existují i droppery, které se účastní i pozdějších fází šíření viru.

10. **Injektory (Injectors)**

    Injektory jsou podobné dropperům. Na rozdíl od nich ale malware zavádějí do paměti jako ovladače nebo mohou přepsat tabulku přerušení, aby systém volal škodlivý kód. Často se používají při rozsévání (seeding) virů, tzn. rozšíření neaktivních virů do více systémů. Pomocí injektorů jsou pak tyto viry spuštěny naráz, aby vypukla rozsáhlá epidemie a bylo těžší vystopovat původce.

11. **Auto-Rootery (Auto-Rooters)**

    Využívají se na testování cílů. Obvykle to jsou skripty a kolekce programů, které odhalují bezpečnostní díry a zaznamenávají je pro pozdější použití. Dají se ale použít i pro infikování právě objevené díry.

12. **Kity – generátory virů (Kits — Virus Generators)**

    Takovýto generátor obvykle vytváří všechny nové viry navzájem podobné, proto není těžké je detekovat. Tímto způsobem viry generují převážně začátečníci, jelikož lze v kitu nastavit požadované vlastnosti viru i bez znalostí programovacích jazyků. Výsledkem generování je zdrojový kód, který poté stačí pouze zkompilovat, čímž získáme vir.

13. **Programy pro spam (Spammer Programs)**

    Spam je nevyžádané masově šířené sdělení (nejčastěji reklamní) rozšiřované po internetu. Toto označení bylo přijato nejprve pro mnohonásobné rozesílání téže zprávy na Usenetu. Následně se ale význam posunul k šíření různých nepřípadných textů a reklamy. Zachoval se i poté, co tento fenomén postihl i ostatní druhy internetové komunikace – např. diskuzní fóra, komentáře nebo instant messaging. Současně se spamem se rozšířil i phishing. (Phishing je podvodná technika používaná k získávání citlivých údajů. Principem je rozesílání e-mailů, které se tváří jako oficiální zpráva známé organizace. Odkazuje ale na stránku útočníka, která

napodobuje stránku oné organizace, a umožňuje útočníkovi sbírat zadávané informace.) Příkladem může být došlá zpráva, že si máte změnit heslo v bance, s adresou, kde to lze provést. Útočníkovi pak už jen stačí si takto získaná hesla ze svých stránek ukládat.

14. **Floodery (Flooders)**

    Tento program generuje nadměrný síťový provoz na konkrétní IP adresu/adresy, což vede k DoS (z anglických slov denial of service, odmítnutí služby). Takto zahlcené počítače nestíhají zpracovávat síťový přenos a nereagují ani na regulární dotazy. Pokud se útočí zároveň z více počítačů, nazýváme takový útok distribuovaný DoS (DdoS).

15. **Rootkity (Rootkits)**

    Rootkity představují speciální sadu pomůcek. Používají se po získání přístupu do systému (třeba po použití nějakého exploitu), aby zajistily útočníkovi přístup do tohoto systému podle toho, jak si bude přát. Mohou obsahovat speciálně upravené aplikace nebo části jádra, které útočník zamění a umožní mu se kdykoliv přihlašovat, cokoliv logovat nebo zakrývat jeho činnost.

16. **Zábavné programy (Joke Programs)**

    „Zda-li má být program klasifikován jako zábavný program nebo trojský kůň, velmi závisí na smyslu pro humor napadeného uživatele.“, A. Solomon

    Tyto programy nemusí být nutně škodlivé, často ale uživatele ruší. v některých případech avšak škodu způsobit mohou.

    Když kamarád někomu spustí na počítači program a on pak neví, jak ho vypnout, může restartováním počítače přijít o neuložená data nebo se tím dokonce může poškodit souborový systém.

    Mezi další podobné programy patří hýbátko myší, změněný zaheslovaný spořič nebo různé náhodně vylézající postavičky.

17. **Poplašné zprávy: řetězové dopisy (Hoaxes: Chain Letters)**

    Hoax (anglické slovo hoax označuje podvod, mystifikaci či žert) je nevyžádaná zpráva, která uživatele varuje před virem, prosí o pomoc, informuje o nebezpečí, snaží se ho pobavit apod. Hoax většinou obsahuje i výzvu žádající další rozeslání hoaxu mezi přátele, případně na co

největší množství dalších adres. Proto se někdy označuje také jako řetězový e-mail. Ač se to nezdá, tyto zprávy mohou způsobit veliké škody.

Důvody nebezpečnosti hoaxů [2]

- obtěžování ostatních

  Opakované čtení stále stejné zprávy nebo nevyžádaná zpráva, která dotyčného nezajímá, rozhodně příjemce nepotěší.
- nebezpečné rady

  Některé hoaxy radí, jak odstranit domělý malware, ale přitom mohou poškodit systém nebo zapříčinit ztrátu důležitých dat.
- zatěžování linek a serverů

  Pokud takový email odešlete pěti lidem a každý další také pěti, tak v desátém kroku email přijde desíti milionům lidí. Většinou se takovýto e-mail přeposílá a nechává se seznam lidí, kteří ho poslali, ten po čase může narůst do obřích velikostí.
- ztráta důvěryhodnosti

  Odesílatel nepravdivých zpráv ohrožuje svoji důvěryhodnost. Může ohrozit i firmu či úřad, pokud využívá pracovní e-mail.
- prozrazení důvěryhodných informací

  Už jenom samotné přeposílání s ponecháváním adres předchozích odesílatelů nahrává odesílatelům spamu. Mnoho hoaxů ale vyžaduje i třeba zadání rodného čísla nebo adresy, případně vyplnění rozličných dotazníků.

18. Adware, Spyware

Adware (advertising-supported software) označuje produkty znepříjemňující práci zobrazováním reklamy, od běžných bannerů až po neustále vyskakující pop-up okna nebo ikony v oznamovací oblasti. Další nepříjemnou věcí je např. změna domovské stránky ve webovém prohlížeči, aniž by o to uživatel měl zájem.

Spywarem nazýváme program, který využívá internetu k odesílání dat z počítače bez vědomí jeho uživatele. Na rozdíl od backdooru jsou odcizována pouze data typu přehled navštívených stránek či nainstalovaných programů. Tato činnost bývá odůvodňována snahou zjistit potřeby nebo zájmy uživatele a tyto informace využít pro cílenou reklamu. Nikdo však nedokáže zaručit, že informace nebo tato technologie nemůže být zneužita. Důležitým poznatkem je, že spyware se šíří společně s řadou sharewarových programů. Jejich autoři o této skutečnosti vědí. Jakmile si program nainstalujete a spustíte, nainstaluje se do systému také spyware.

Rozdíl mezi pojmy spyware a adware spočívá v tom, že adware velmi často využívá výsledků, které dokázal vyprodukovat spyware, ale není na něm závislý. Adware se instaluje do počítače za souhlasu uživatele. Naproti tomu spyware se instaluje do počítače bez vědomí a souhlasu uživatele.

Schéma pojmenování malware

Tento způsob pojmenování vznikl ze stejného důvodu jako doménové jméno k IP adrese. Lidé si totiž obecně lépe pamatují jméno v textové podobě než v číselné.

Následující zápis je v nejkomplexnější podobě:
<malware_type>://<platform>/<family_name>.<group_name>.<infective_length>
.<variant><modifiers>

Překlad by byl následující:
<typ škodlivého software>://<platforma>/<jméno rodiny>.<jméno skupiny>.<infekční délka>.<varianta><modifikátory>

Obecně se však nevyužívají všechny kolonky. Často se používá pouze jméno rodiny, někdy se přidává, o jaký typ malware jde, pro které platformy je určen a pokud vzniklo více variant, tak se uvedou i ty. Mezi modifikátory pak patří další přesnější určení malware, například omezení konkrétní lokalizace hostitele. (Například dřívější makroviry se šířily jenom pod anglickou verzí Microsoft Word, lokalizované verze měly přeloženy i samotné názvy funkcí. Tyto viry pod nimi nebyly provozuschopné, protože volaly funkce anglickými jmény.) Dalším modifikátorem může být použitá komprimace spustitelného souboru, například, zda jde o UPX nebo další. Případně upřesnění, zda se vir šíří e-mailem „@m“ (z anglického Mailer) nebo se rozesílá ve velkém „@mm“(Mass-mailer).

3 Stručný přehled vývoje malware

V této kapitole bych nejprve rád popsal časovou osu. Budu se snažit poukázat na jednotlivé kroky vývoje malware. Poté zdůrazním jednotlivé vlastnosti těchto programů a uvedu jejich nové formy.

Evoluce malware se děje jako série skoků oddělených dlouhými periodami statického klidu. Již po naprogramování prvního viru byl vytvořen i první antivir. Od té doby mezi sebou soutěží.

3.1 Časová osa

1971-1972

Vznikl program Creeper, jeden z prvních virů, který byl následován prvním antivirem Reaper. Oba vytvořil stejný autor.

1975

ANIMAL byla infekční hra pro UNIVAC, jež se šířila na děrných páskách.

1982

Elk Cloner – virus pro Apple II, tedy první pro mikropočítače.

1984

Dr. Frederic Cohen použil poprvé termín počítačový virus a matematicky ho definoval.

1986

Brain byl první boot virus, který zároveň jako první používal stealth techniky.

listopad 1988

V listopadu 1988 napadl Morrisův červ počítače unixové koncepce. K šíření využíval **bezpečnostní chyby** v návrhu protokolu SMTP a **přetečení bufferu.** Zkoušel se také přihlásit na účty s **lehce uhodnutelnými hesly**. Na napadených počítačích používal maskovací metody (**stealth**). Po síti bylo jeho tělo přenášeno v **šifrované podobě**. Červ napadl kolem 6000 počítačů, škody dosáhly 100 miliónů dolarů. Jeho autor, Robert Tappan Morris, tvrdil, že se mu vymkl z kontroly.

1990

První polymorfní vir se jmenoval 1260.

1991

MtE (Dark Avengerův mutovací engine) – část kódu, která se dá připojit k vlastnímu viru a po zavolání kolem něj vytvoří polymorfní obal.

květen 1994

OneHalf – ve své době velmi rozšířený polymorfní virus, který šifroval data na disku. Používal symetrické šifrování, takže klíč k dešifrování měl stále u sebe. Když systém/uživatel přistupoval k šifrovaným datům, tak je vir dešifroval a poslal je zpět čitelné. Celé tělo viru bylo náhodně rozptýleno po infikovaném souboru. Části, které přepsal, pak byly přidány na jeho konec. Aby se nepoznalo, že soubor byl změněn, zajišťoval vir správnou funkčnost přepsaných částí.

srpen 1995

Concept se stal prvním makrovirem, šířil se výhradně v MS Word.

leden 1996

W95/Boza.A napadal PE soubory a jako první používal WinAPI funkce.

1996

W95/Punch byl první paměťově rezidentní virus pro Windows 95, přežíval jako VxD ovladač.

červen 1998

Win95/CIH – „Černobyl“ je mediálně známý jako vir, který dokáže zničit hardware. Dokázal přepsat Flash BIOS uložený na základní desce, obvykle to ale nebylo nezvratné.

1998

W95/HPS, W95/Marburg – s rozšiřováním systémů od firmy Microsoft se objevují i plně 32bitové polymorfní viry.

1998

W95/Regswap – protože polymorfismus na ochranu viru nestačil, objevily se metamorfní viry. Tento měnil používané registry editací svého binárního kódu.

Příklad: 8BF5 {mov esi, ebp}
8BD5 {mov edx, ebp}

1999

W97M/Mellisa – se zvýšením počtu uživatelů internetu nastává rozmach používání e-mailů a i šíření malware. Příkladem může být právě tento makrovirus, který se šířil v dokumentech Wordu.

2000

W95/Zmist byl metamorfní virus. Náhodně používal dekryptor a jako první přišel s integrací kódu do infikovaných programů.

2002

W95/Sma byl první stealth virus určený pro Windows 95/98.

červenec 2002

Červ Linux/Slapper se stal snad nejrozšířenějším malwarem v Linux-like systémech. Tento navíc vytváří P2P sítě z napadených stanic. Napadené počítače pak mohou být ovládány na dálku a často slouží k DDoS (distribuované odmítnutí služby) útoku.

leden 2003

Červ W32/Slammer předeběhl všechny dřívější škodlivé programy v rychlosti šíření.

2004

W64/Rugrat.3344 je plně 64bitový vir pro Windows. Protože starý formát PE nebyl pro 64bité aplikace vhodný, musel být vymyšlen nový, PE+.

3.2 Přizpůsobení malware prostředí

Pro každou platformu i operační systém již vznikl alespoň jeden malware. První z těch známějších se nazývá „Morrisův červ“. Vše, co využíval pro svoji existenci a šíření, používá většina škodlivého kódu i dnes (od obyčejného zneužití bezpečnostní díry po nízké zabezpečení). Při každém přechodu na novou platformu či nový systém se vývoj nevrací zpět, pouze chvíli stagnuje, než si programátoři zvyknou na nové možnosti a plně je využijí.

Významné pro vývoj malware byl příchod DOSu. Především proto, že získal vysokou popularitu a rychle se rozšířil. Společně s viry ale vznikaly i antiviry. Pro zvýšení své životnosti přišly viry s oligomorfismem a následně i polymorfismem.

Oligomorfismus znamená použití více dekryptorů, které jsou nošeny v těle viru. Náhodně se vybere, který bude použit pro samotné zašifrování viru. Pro další instanci se aplikuje jiný. Jejich počet je pevně dán.

U polymorfních virů závisí počet dekryptorů na množství kombinací instrukcí, z kterých je požadovaný dekryptor složen. Může jich být až několik miliónů.

Mezi další pasivní vlastnosti patří i vyhýbávání se souborům podezřelých z toho, že by mohly patřit nějakému antiviru. Některý malware svoji bezpečnost řeší hrubou sílou a snaží se antiviry a firewally vypínat nebo přímo mazat z disků. Tyto metody jsou nadále využívány a neustále vylepšovány.

S nástupem operačního systému Windows 95 byl slibován zánik virů. První vir však byl distribuován na disketách spolu s beta verzí samotného systému. Současně se objevilo i nové prostředí, dostatečně rozšířené a zároveň programovatelné – Microsoft Office – a na scénu vstoupily makroviry.

Internet je další z prostředí, které napomáhá šíření malware. Zároveň ho může ale i omezovat, například větší informovaností uživatelů nebo lehčeji a mnohonásobně rychleji aktualizovatelnými antiviry. V nekončícím boji s antiviry objevily viry metamorfismus.

Metamorfismus představuje schopnost vypadat po každém sestavení jinak, a to bez použití dekryptoru. První viry začínaly s pouhou výměnou registrů a přímým přepisem binárního kódu. Pozdější viry byly rozděleny na několik úseků kódu a jejich pořadí bylo při vytváření nové instance zpřeházeno tak, aby zůstala funkčnost. Případně bylo tělo doplněno o různé „smetí“, které chod nijak neovlivní, jenom změní velikost a podobu celého viru.

V roce 1991 bylo ve světě známo kolem 300 různých virů. O čtrnáct let později, v roce 2005, jich bylo už 140 000. V těchto číslech nejsou započítány různé varianty jednotlivých virů a ani malware, který není antiviry detekován. [13]

Z této části mé práce je patrné, že se tyto programy postupně vyvíjejí. Vždy když se objeví nová myšlenka nebo nové prostředí, tak se malware po čase přizpůsobí. Využívají starších osvědčených metod a kombinují je s novými. Dnes se zaměřují především na bezpečnostní díry a sociální inženýrství.

4 Zajímavý malware

V této kapitole popíši několik virů a červů, které si myslím, že stojí za povšimnutí. Hned první ukázka se týká kryptovirologie (více v podkapitole 6.5). Druhým příkladem je červ, který šikovně využívá aktualizaci kódu a rozšiřování funkčnosti. Třetí podkapitola se týká červa vytvářejícího botnet (opět více později, v podkapitole 6.1). Poslední pak popisuje vir používající poměrně novou techniku – integraci kódu – spojenou s polymorfismem, metamorfismem a s velkým množstvím náhodného rozhodování.

4.1 OneHalf

OneHalf je dosový polymorfní virus. Napadal boot sektor disku a infikoval i soubory COM a EXE. Je znám především kvůli svému šifrování. Při svém spuštění OneHalf vždy zašifroval část dat disku. Pokud byl ale zároveň aktivní, tak za letu dešifroval ty soubory, ke kterým uživatel přistupoval. Tím zakrýval svou přítomnost, aby mohl poklidně pokračovat ve své činnosti. Nenapadal soubory s následujícími názvy SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV, CHKDSK, což představovalo ochranu před odhalením. Tyto názvy souborů totiž používaly tehdejší antiviry.

Po zašifrování poloviny disku vypsal následující zprávu:

Dis is one half.
Press any key to continue ...

Pokud byl virus neopatrně odstraněn, nemuselo se podařit zachránit jím zašifrovaná data. Slabinou symetrického šifrování, které se zde používalo, bylo šifrování i dešifrování jedním a tím samým klíčem. OneHalf ho ukládal na disku v partition tabulce. Stačilo ho tedy objevit a záchrana dat byla bezproblémová.

4.2 W95/Hybris (září 2000)

Tento červ je výjimečný hned z několika důvodů. Na jeho vývoji se podílelo více osob, dokonce by se dalo říci špičkových tvůrců virů z celého světa. Hlavní autor vystupuje pod přezdívkou Vecna (z Brazílie, člen skupiny 29A). Při tvorbě vycházel ze svého předešlého červa W95/Babylonia (červenec 1999), který měl slabinu ve způsobu aktualizace. Původní červ infikoval soubory nápovědy systému Windows a soubory PE. Stahoval si aktualizace z webového serveru. Tato aktualizace

probíhala ve formě plug-inů, díky kterým šel celý červ vylepšit a mohly se přidávat nové funkce. Ona slabina spočívala v tom, že byl jenom jeden server, odkud se aktualizace stahovaly. Po jeho zablokování se červ už nevyvíjel. Navíc neověřoval stažené plug-iny, takže je bylo lehké podvrhnout. (více viz Válka červů).

Hybris používá 1023bitový RSA podpis a 128bitovou hashovací funkci pro ochranu aktualizací před případným útokem. Tato hashovací funkce používá XTEA, tzn. rozšířený kódovací algoritmus, který je následníkem algoritmu TEA. XTEA představuje program typu public domain vytvořený D. Wheelerem a R. Needhamem. RSA knihovnu pro červa vytvořil ruský programátor s přezdívkou Zombie (viz Zmist). Aktualizace byla zašifrována pomocí algoritmu XTEA a podepsána pomocí RSA na útočníkově systému. Probíhalo to tak, že útočník vytvořil tajný klíč a k němu odpovídající veřejný klíč. Veřejný klíč se vložil do červa, přičemž kódovací a dekódovací klíče XTEA podepsané tajným RSA klíčem byly doručeny společně s modulem. Přestože byly aktualizace zašifrovány, jednalo se o algoritmus používající symetrický klíč. Moduly mohly být tedy dekódovány podobným způsobem, jakým je dekódoval samotný červ. Útočník byl však chráněn proti manipulacím, které by měly za cíl úpravu modulů.

Pro Hybris bylo známo přes dvacet různých modulů a více než třicet dva různých verzí. Přestože původní aktualizační webová stránka byla velmi rychle odstraněna, útočníci měli možnost zasílat nové aktualizace prostřednictvím newsgroups. Infikované uzly pak posílaly moduly zpátky do těchto newsgroups, takže všechny infikované uzly měly možnost dostat se k aktualizacím.

Níže uvádím různé moduly, které červ může využívat:

- Modul napadající EXE soubory systému DOS
- Modul určený pro infekci PE souborů takovým způsobem, aby se nezměnila ani jejich velikost ani kontrolní součet CRC 16/32/48. Tento modul používá kompresi pro zkomprimování hostitele a zaplnění modulu dalšími daty. Pomocí algoritmu, vytvořeného ruským autorem virů Zhengxi, bylo možno dosáhnout stejného kontrolního součtu jako před infekcí.
- Modul pro zakódování souboru WSOCK32.DLL infikovaného Hybrisem. Toto umožňovalo červu šířit se i pomocí e-mailů. Technika je převzata z červa Happy99. Červ zapíše část svého kódu za funkce *connect, recv a send*, aby si zajistil spuštění ve správnou chvíli.
- Modul pro infekci souborů nápovědy Windows
- Modul pro infekci používající polymorfní funkce KME (Kewl Mutation Engine) autora Zombie. Byl prvně použit u viru W95/ZMorf.

- Dva moduly umožňující infekci uvnitř archivů RAR, ZIP a ARJ
- Dva rozdílné moduly pro infekci dokumentů Microsoft Word a Excel
- Modul pro DoS útoky
- Zakódovaný generátor trojských koní
- Modul umožňující infekci pomocí zadních vrátek viru SubSeven
- Modul zpráv HATE (human-alike text engine). Tento modul je schopen generovat zprávy se jmény známých antivirových výzkumníků. Umožňuje odesílání e-mailových zpráv za použití jedné z adres v poli odesílatele. Tvůrcem je španělský autor virů Mr. Sandman, zakladatel skupiny 29A, sdružující různé autory virů. Členy byly i dva Češi vystupující pod přezdívkami Benny a Ratter.
- Útočný retro modul bránící v přístupu ke stránkám s antiviry
- Generátor e-mailových zpráv, který za použití webového serveru SOAP generuje přáníčkové zprávy a tyto pak odesílá příjemcům (i s červem Hybris).
- Aplikace infikující soubory SYS za účelem ukrytí infikovaného WSOCK32.DLL pomocí stealth rutin
- Exploitační modul určený k získávání souborů ze zranitelných webových serverů
- Modul určený k prohledávání harddisku a registrů, nalezení antivirových programů a jejich následné smazání, včetně zničení databází
- E-mailově založený modul trackeru posílající zprávy z infikovaných uzlů na určité adresy
- Několik dalších generátorů zpráv určených pro šíření e-maily
- Modul Happy 2000. Přepisuje soubor SKA.EXE červa Happy99, který pak šíří Hybris. Obsahuje rovněž grafický payload červa Happy99.
- Modul pro stahování dodatečných plug-inů z webových stránek
- Usenetový modul pro připojování se k serverům NNTP a stahování plug-inů. Tento modul také uploaduje jiné moduly do newsgroups. Používá seznam s více než 70 různými adresami.
- Animace založená na OpenGL, která se nainstaluje při bootování. Tento modul vytvořil francouzský autor virů Spanska.

4.3 Linux/Slapper (2002)

Slapper, linuxový červ, zneužívá chyby objevené 30. července 2002 v balíku OpenSSL, který je využíván pro implementaci protokolu SSL (Secure Socket Layer). OpenSSL se používá u mnoha populárních softwarových balíků, ale především ve webovém serveru Apache.

Před samotným útokem červ nejprve zjistí informace o potencionální oběti – zasláním chybného požadavku GET na HTTP port (port 80). Při defaultním nastavení serveru se odešle zpět chybový stav a společně s ním i číslo verze serveru Apache a verze distribuce Linuxu, na kterém běží. Červ obsahuje pevný seznam 23 architektur, na kterých byl testován. s tímto seznamem porovná získané informace. Podle srovnání pak upravuje postup útoku. Pokud server nevrací číslo verze a architekturu, červ předpokládá, že to je Apache 1.3.23 běžící na RedHatu.

Mechanismus útoku je založen na dvojnásobném využití přetečení haldy (anglicky heap). Vždy jde o chybu v kontrole velikosti přijatých dat. Při zpracovávání těchto dat je server nakopíruje do bufferu pevné délky, ale jelikož dat je více, je přepsána i paměť následující po vyhrazeném místě. Červ se tímto způsobem dostane do systému. Druhé přetečení pak zajistí jeho spuštění, tedy jen jeho části ve formě shellskriptu, která zkompiluje vir a spustí ho. Vše je prováděno ve složce „/tmp“, kde nejsou pro zápis vyžadována práva administrátora.

Samotné přenesení červa na hostitelský systém se provádí bez využití šifrovaného spojení, protože k útoku na buffer dochází příliš brzy. Proto není složité červa detekovat v paketech, kde se jak shellskript tak jeho tělo šíří v texové podobě.

Po spuštění se vytvoří připojení k portu 2002/UDP a červ se pokusí spojit s P2P (peer-to-peer; označení síťové architektury, kde klienti komunikují bez zprostředkovatele/serveru přímo mezi sebou) sítě. Adresu infikovaného systému červ poté pošle útočníkovi, který ji rozešle náhodné skupině uzlů v P2P síti. Tyto uzly ji postupně rozesílají dál, aby o novém přírůstku věděli i ostatní. Tomuto říkáme segmentová vysílací technika (broadcast segmentation technique). Každý uzel umožňuje dálkové spouštění příkazů a rozličné DoS útoky (UDP flood, TCP SYN flood, IPv6 SYN flood). Protože byl útok v počátku veden z více počítačů, vzniklo několik paralelních sítí. Největší taková nalezená síť měla skoro 20 000 počítačů.

4.4 W95/Zmist (2000)

Na konferenci Virus Bulletin 2000 byly uvedeny výsledky výzkumu dvou odborníků z IBM (Dave Chess a Steve White) na téma nedetekovatelné viry. Po této konferenci vydal ruský autor virů, Zombie, článek „Undetectable Virus Technology“ (Technologie nedetekovatelných virů). Následně se objevil jeho vir Zmist. I dřívější viry tohoto autora byly známy svou složitostí a propracovaností, zde však zvedl laťku ještě výše.

Vir nepoužívá jen běžné metody, tj. metamorfismus, polymorfismus a další ochrany proti odhalení, ale přichází i se zcela unikátní technikou integrace kódu. Jeho engine Mistfall umí dekompilovat PE soubory, na což potřebuje 32 MB paměti. Poté se Zmist vloží do kódu hostitele, přesune bloky kódu jinam, vloží sám sebe, zregeneruje kód a data (včetně relokačních informací) a znovu sestaví spustitelný soubor. Za každou instrukci v sekci kódu navíc Zmist vkládá instrukci skoku vedoucí na další instrukci. Takto modifikovaný program bezchybně fungoval, čemuž se i sám autor divil.

„V praxi jsme nezaznamenali jediný pád aplikace způsobený během replikace.“ [1]

Virus se tímto způsobem stane částí toku kódu. Jelikož se umísťuje na náhodné místo, je možné, že se nikdy ani nespustí. Pokud se aktivuje, okamžitě spustí hostitele jako separátní proces a ukryje se. Mezitím virus vyhledává vhodné soubory pro další infekci. Samotné sestavení (zmutování) viru se na počítači provádí pouze jednou, a to z důvodu vysoké časové a paměťové náročnosti. Využívá se engine RPME (Zombie ho již použil u viru W95/Zperm). Náhodnost se aplikuje na většinu rozhodování. Je pravděpodobné, že nalezený soubor, ač je vhodný a splňuje všechny podmínky, nebude infikován. Je dokonce možné, že v průběhu infikování bude vše přerušeno a vráceno do původního stavu. S pravděpodobností jedna ku desíti se za každou instrukci vkládá instrukce skoku. Kód se vkládá do náhodných oblastí, stejně tak je náhodně použit dekryptor. Existuje zde možnost, že tělo viru bude do souboru přiloženo v nešifrované podobě. Pokud se použije šifrování, kód viru se zakóduje pouze pomocí instrukcí ADD, SUB nebo XOR s náhodným klíčem, který se mění každým průchodem další operace s druhým náhodným klíčem, přičemž mezi instrukce dekódování se vloží různé instrukce smetí. Generují se pomocí Zombieho engine ETG (Executable Trash Generator).

5 Sociální inženýrství

Pojmem „sociální inženýrství" (sociotechnika) označujeme přesvědčování a ovlivňování lidí s cílem oklamat je tak, aby uvěřili, že jste někdo jiný, a mohli být zmanipulováni k vyzrazení některých informací nebo provedení určitých úkonů.

Proč se obtěžovat s používáním brutální síly na prolamování hesel, když jednodušší je přinutit někoho, kdo heslo zná, k tomu, aby nám jej řekl? Navíc při dobře vedeném útoku si oběť v drtivé většině vůbec neuvědomí, že něco vyzradila nepovolané osobě.

Podle sociologie existuje šest základních rysů lidské povahy, které lze lehce zneužít pomocí sociálního inženýrství. Jsou to [6]:

1. **Autorita** – Tendence podřídit se člověku s vyšší autoritou. Nejen výše postavenému ale i vystupujícímu rozhodně. Sem by se daly zařadit hoaxy nebo i červy, které jsou poslané jakoby od IT odborníků nebo známých lidí.
2. **Sympatie** – Lidé častěji pomohou někomu, kdo se jim líbí nebo kdo má podobné zájmy či názory.
3. **Vzájemnost** – Pokud sociotechnik oběti pomůže s nějakým problémem, tak ona na oplátku také bude schůdnější k jeho návrhům.
4. **Důslednost** – Lidé mají tendenci se podřídit, jestliže předtím veřejně vyhlásili svou podporu a angažovanost v určité záležitosti.
5. **Společenský souhlas** – Všichni s tím souhlasí, tak proč by člověk měl jít proti ostatním? Sem by se určitě řadily e-maily žádající o další rozeslání. Chcete, aby někdo vyplnil dotazník? Připište na něj, že všichni ostatní ho už vyplnili a že dotyčný je poslední. Pravděpodobnost vyplnění pak dosahuje skoro 100 %.
6. **Vzácná příležitost** – Na toto spoléhá většina reklam se slevami nebo omezenou nabídkou. Vzácná příležitost může však být i zneužita k získávání informací.

Třetí tisíciletí je nazýváno informačním věkem. Stále častěji můžeme slyšet či se dokonce sami přesvědčit, že úspěšný bude ten, kdo bude ovládat schopnost získávat, hledat a správně vyhodnocovat informace. Lidé si však stále ještě neuvědomují jejich hodnotu. Málokoho napadne, že informace je také nutno odpovídajícím způsobem střežit.

Na rozdíl od člověka může malware využívat pouze předem naprogramované metody útoku. To ho sice omezuje, ale vynahrazuje si to paralelním útokem na více cílů – geograficky, věkem i vzděláním vzdálených. Člověk si může dopředu připravit různé strategie a velice pružně je přizpůsobovat potřebě. Malware se snaží spíš využívat neznalost a nepodezíravost napadených. Mnoho škodlivých programů by se nešířilo, kdyby každý uživatel kontroloval nové soubory pomocí aktualizovaného antiviru, případně si ověřoval u odesílatele, zda je zpráva opravdu od něho.

Do současnosti malware pro své šíření používal bezpečnostní díry v programech a systémech. Další možností bylo nechat se podvodem spustit. Tyto programy pak byly odchytávány a blokovány antiviry, které jsou nyní velmi komplexní a díky rychlé aktualizaci přes internet i velmi výkonné.
Podle studie A pack with the Devil, jejíž autoři jsou Mike Bond a George Danezis, na nás kouká malware jako na nepřítele. Ale co když začne chytře využívat sociální inženýrství v plném rozsahu toho, co nabízí? Co kdyby nabídl něco na oplátku lidem, kteří ho nainstalují? Co když to bude tak lákavé, že nikdo neodmítne? Nebylo by lepší, aby uživatel přistoupil ke spuštění dobrovolně? Malware si pak postupně vytvoří pouta a uživatel ze závislosti na něm jen tak neunikne. Ke spolupráci může uživatele donutit jeho chtivost, zvědavost, potřeba mít převahu nebo strach. Malware ho může uplácet informacemi, které nelze odmítnout. Tudíž na scénu přichází spolupráce s uživateli.

Tato interakce by se mohla skládat například z těchto bodů [7]:

1. **Pokušení**

   Tento program se dokáže šířit i obvyklými cestami, což je stále jeho primární šíření v počáteční fázi, než začne využívat sociálního inženýrství. Mějme tři lidi: Alici, Boba a Evu. Alice je už napadená, malware ji sleduje a najde u ní kontakt na Boba. Pošle mu e-mail, ve kterém nabízí přístup k jejím dokumentům a e-mailům. Aby takto nahozený háček byl ještě lákavější, přidá i ukázku toho, co může nabídnout. Bob má na výběr, buď dobrovolně přistoupí a nainstaluje tento malware u sebe. Zdůvodněním může být, že získá vyhledávací rozhraní pro vyhledávání v Aliciných dokumentech. Nebo může odmítnout a dokonce i Alici oznámit, že ho má na počítači.

   Mezi další lákadla můžeme zařadit možnost monitorovat práci s počítačem u podřízených (nebo u dětí, případně i partnera), kdy pak takový škodlivý program bude distribuován ve formě klienta a serveru. Dále bych uvedl program umožňující připojení do P2P sítě, kde je sdílena hudba nebo filmy a podobná autorským zákonem chráněná díla. Čím více bude napadeno počítačů, tím větší bude počet sdílených souborů a tím větší bude zájem se do sítě

připojit. Tyto dva případy se už využívají, ale zatím ne takto komplexně. Zůstává se pouze u sledování z důvodu cílenější reklamy, různých statistik nebo posílání spamu. Tyto programy řadíme mezi spyware.

2. **Monitorování**

   Po nainstalování program začne monitorovat i Boba, vše co dělá, i to, co vyhledává u Alice. Kdyby Bob program zkoušel odstranit, vše se ukládá a zálohuje u Alice nebo dalších napadených. Napadené počítače spolu musí nějak komunikovat. Řešením je třeba P2P síť, kterou lze využít i pro výměnu dat či sdílených souborů.

   Jelikož se ukládá vše, co Bob hledal, nemusí se řešit zjišťování zajímavého materiálu, Bob to dělá sám. Tato data se pak dají využít v dalším lákání potencionálních cílů nebo při vydírání Boba.

3. **Vydírání**

   Když je nasbíráno dostatek informací, zašle se nejdříve varování Bobovi, co vše může být zveřejněno proti němu v případě, kdyby vir odstranil. Opět podpořeno ukázkou nasbíraných dat ze sledování. To, že je vše uloženo na jiném počítači, umožňuje zveřejnit tyto informace, i přesto, že Bobův program přestane odpovídat (byl odstraněn). Varování lze využít i k získávání podstatných informací, pokud Bob nemá čisté svědomí a chce něco schovat. Po přečtení varování začne mazat inkriminované soubory. Malware je poté pouze schová a může je později využít. V opačném případě, pokud Bob nemá co skrývat a ani zveřejnění mu nevadí, začne služby malware využívat ještě víc.

4. **Dobrovolné šíření**

   Po fázi vydírání může přijít fáze šíření. Bob už ví, že mu hrozí nebezpečí, a je tedy připraven pomáhat. Malware se zeptá na cíl a Bob vybere Evu. Bob buď jeho novou instanci přímo nainstaluje na počítač nebo jenom poskytne kontakt na osobu, která by mohla mít také zájem o Alicina data. Po úspěšném rozšíření je pak Bob odměněn i možností sledovat Evu.

5. **Nedobrovolné šíření**

   Když je shledáno, že Bob neplní svoji část dohody, tak se malware rozešle na všechny Bobovy kontakty (nasbírané v kontakt listu, v jeho dokumentech a podobně) a adresátům slíbí, že budou mít přístup k Bobovým dokumentům. Lákavost může zvýšit přidáním ukázky.

6 Hlavní směry vývoje malware

Následující tři podkapitoly pojednávají o současném malware. Nejméně známým, ale zároveň nejrozšířenějším malwarem jsou boty přispojené do botnetu. Mezi mediálně nejznámější patří tzv. „dobré viry“ (ve skutečnosti to jsou všechno červi). Dále uvádím distribuované výpočty, algoritmy, které zatím nejsou malwarem příliš používané, ale je jisté, že se brzy rozšíří. Ve čtvrté podkapitole popisuji kryptovirologii a v poslední uvádím hlavní rysy, které by měl splňovat budoucí malware, aby byl uspěšný.

6.1 Botnet

Botnet představuje virtuální síť složenou z botů. Botem je myšlen program, který je nainstalován na napadeném počítači a umožňuje útočníkovi ovládat takový počítač na dálku z jiného počítače. Samotný termín bot pochází ze slova českého původu „robot“ a stejně jako robot je využíván pro různé „práce“.

Díky možnosti vzdáleného ovládání je bot velmi flexibilní. Útočník může tento program vylepšovat o nové funkce nebo jenom měnit stávající. Takto ovládané počítače jsou nazývány „zombie“, jsou jim totiž podobné, také jsou bezduché a ovládané někým jiným.

Boty mohou být kontrolovány několika způsoby. Dříve zmíněný Hybris využíval diskuzní skupiny (newsgroups). Dále se používají P2P sítě nebo IRC servery.

IRC (Internet Relay Chat) je protokol, který zprostředkovává komunikaci klientů mezi sebou. Byl to jeden z prvních protokolů umožňující komunikovat v reálném čase. Pochází z roku 1988 a ve své době byl velmi rozšířený. Běžní uživatelé dnes převážně využívají jiné protokoly, například ICQ, jabber nebo MSN, ale IRC má stále mnoho příznivců. Mezi hlavní důvody jeho používání patří možnost konfigurovatelnosti přes skripty; větší výběr serverů, ke kterým se dá připojit; vyšší anonymita (komunikace může být přenášena přes server, takže klient nemusí znát IP adresu ostatních) a bezpečnost (komunikace se dá šifrovat přes protokol SSL). Dále je výhodou, že díky svému stáří je dostatečně otestován vůči různým hrozbám, jako jsou bezpečnostní díry v serverech či chyby v návrhu samotného protokolu, a má už hotové knihovny, které jsou volně dostupné.

Hlavní nevýhodou komunikace přes IRC je jeho centralizace. Je sice nepravděpodobné, že by si někdo dovolil odstavit server s padesáti tisíci lidmi a pak jim toto odpojení zdůvodňoval, ale záleželo

by zde na velikosti hrozby vyplývající z komunikace těchto botů. Navíc pokud bude mít bot seznam více serverů, na které se bude připojovat, a pokud některý nebude dostupný, tak se vypnutím jednoho z nich nic nevyřeší.

P2P (peer-to-peer) zde není žádnou konkrétní sítí. Sice existuje několik volně dostupných implementací, které útočníci mohou použít, ale většinou si navrhnou vlastní síť přímo na míru. Šifrovaná komunikace a elektronický podpis jsou dnes nutností. Výhodou je, že bude umět všechno, co vyžadují, a pro antivirové společnosti i ostatní bude mnohem obtížnější ji pochopit a bojovat proti ní. Nevýhodu pak představuje nutnost ji naprogramovat a navrhnout, což může být dost obtížné, zejména má-li být velmi komplexní. Dalším záporem je možnost vytvoření chyb přímo v programu a návrhu. Takováto síť je totiž obtížně testovatelná, takže zde existuje vysoká pravděpodobnost, že se chybám nelze vyhnout.

Dle odhadu expertů ovládají útočníci 7 % z celkového počtu počítačů na celém světě, to činí přibližně 47 milionů strojů. Za vše hovoří případ z praxe, kdy v říjnu 2005 byli v Holandsku zatčeni tři muži, kteří měli k dispozici botnetovou síť obsahující přes 1 500 000 počítačů. [12]

Dle zprávy společnosti Symantec o internetových hrozbách v druhé polovině roku 2005 patřilo mezi země nejvíce infikované botem USA (26 %), Velká Británie (22 %) a Čína (9 %). Průměrný počet infikovaných počítačů botem byl kolem 9163 za den. [12]

Nejčastěji ale botnet slouží k šíření spamu (dle skupiny Gartner Group je 70 % spamu rozesíláno právě z botnetů). Dalším využitím může být provádění DDoS útoků, získávání informací z více zdrojů najednou (hesla, e-mailové adresy, dokumenty) nebo k vypouštění dalšího nového malware. [12]

6.2 Válka červů

- Linux/Lion versus Linux/Cheese
  Lion si nechával na napadených počítačích zadní vrátka, ty Cheese odstraňoval. Poté se šířil na další počítače kompromitované Lionem.

- W32/Welchia versus W32/Blaster
  Welchia a Blaster napadali stejnou chybu v RPC (Remote procedure call) na Microsoft Windows. Welchia patřil mezi „antivirové“ červy, po proniknutí do počítače nainstaloval

záplatu. Problém spočíval v tom, že to udělal bez vědomí uživatele; nelogoval, co všechno provádí a restartoval aktualizovaný počítač. To mohlo způsobit mnoho komplikací, od ztráty neuložených dat až po nefunkčnost systému. Tento červ navíc výrazně zatěžoval síť.

- W32/Sasser versus W32Gaobot
  Sasser využíval stejnou bezpečnostní díru jako Gaobot, jehož tvůrci se to nelíbilo, protože oba červi museli bojovat o volné cíle. Vydal tedy novou verzi Gaobota, která používala upíří techniku. Místo aby Sassera odstranil, pouze ho pozměnil, aby místo sebe šířil Gaobota. Sasser pak normálně napadal okolní počítače, ale když se měl nainstalovat do potencionálního cíle, nainstaloval Gaobota.

  A aby toho nebylo málo, Sasser si po napadení nového počítače stahoval vlastní tělo z jednoduchého FTP serveru, který obsahoval chybu – jednoduché přetečení bufferu, které využíval červ W32/Dabber. Byl to vlastně Dabberův jediný způsob šíření. Vyhledával počítače napadené Sasserem a díky této zranitelnosti se šířil.

- W32/CodeRed versus W32/CodeGreen
  CodeGreen, další z „antivirových" červů, likvidoval infekci červa CodeRed. Zároveň odstraňoval bezpečnostní díru, kterou CodeRed využíval pro napadení počítače, nainstalováním oficiální záplaty. CodeGreen byl uložený pouze v paměti, restartování počítače ho odstranilo.

Tato podkapitola úzce souvisí s předchozí. Výše uvedené příklady bojů mezi různými malware představuje pouze úvod do skutečných válek červů. Reálné boje začaly až mezi červy umožňující vzdáleně ovládat počítače a později mezi červy vytvářející botnet. Jejich cílem byly zdroje (napadnutelné počítače), o které se mezi sebou nechtěli dělit.

První opravdu veliká válka začala na jaře roku 2004, hlavními aktéry byli červi Mydoom, Bagle a NetSky. Tito červi otevírali port, na kterém poslouchali a umožňovali útočníkovi, aby takový počítač mohl ovládat. Samozřejmě, že pokud na napadeném počítači narazili na ostatní červy, tak je neúprosně smazali. Boj přestal ještě téhož roku, protože zatkli autora NetSky. Byl jím 18letý Němec Sven Jaschan, kterého udali jeho kamarádi za 250 000 dolarů. Přiznal se také k autorství červu Sasser. Zajímavostí je, že ještě koncem minulého roku (tedy nejméně po roce a půl od vydání poslední verze) vedl NetSky žebříčky nejrozšířenějšího malware. Jeho autor dnes pracuje pro známou antivirovou společnost. [2]

V současnosti se vede další válka. Účastníky jsou Warezov, Zhelatin a stále Bagle. Podle antivirových odborníků za každým červem stojí celá skupina autorů, každá je z jiné země. Tito červi se snaží vytvořit botnet. Vyznačují se šířením spamu a sbíráním e-mailových adres z napadených počítačů. Všechny tři skupiny to dělají kvůli penězům. Získané adresy prodávají a vydělávají i za odeslaný spam. Navíc se často objevují nabídky „pronajmutí“ botů. Stejně jako předešlí červi mezi sebou bojují a pokud mohou, odstraňují konkurenční nákazu. Dokonce existují varianty, které se snaží napadat celé konkurenční botnety. [14]

6.3 Distribuované výpočty

Distribuované výpočty umožňují složité úlohy rozdělit mezi běžné domácí počítače. Nápad distribuce úloh je starý jako počítače samotné, ale v aktuální podobě je realizovatelný až někdy od devadesátých let, kdy se mezi běžné uživatele dostalo slušné připojení k internetu. Snad nejznámějším projektem tohoto druhu je SETI@home, který pomocí velkého radioteleskopu sbírá signály z vesmíru a hledá v nich známky mimozemské inteligence. [4]

Využití distribuovaných výpočtů malwarem bylo předpovězeno už v roce 1989. Dají se využít například pro lámání šifer a hádání zašifrovaných hesel hrubou sílou. Každý klient dostane určitý rozsah, který má vyzkoušet. Pokud na tom budou pracovat všechny počítače najednou, je možné získat výsledky v brzké době.

Příklady malware, který již tyto výpočty používá:

- W32/Opaserv – Snaží se pomocí distribuovaných výpočtů určit tajný klíč založený na šifře DES.
- W32/Bymer – Funguje jenom jako dropper, který DNETC (Distributed Network Client) stáhne, nainstaluje na napadený počítač a zajistí úpravou registrů jeho spuštění.
- W32/Hyd – Instaluje přímo program SETI @ Home (Search for Extra-Terrestrial Intelligence at home). Tento program pak provádí výpočty nastavené na účtu autora červa.

6.4 Kryptovirologie

Kryptovirologie je věda studující, jakým způsobem by škodlivý software mohl využívat kryptografii (šifrování). Jak bylo řečeno u viru OneHalf, při použití symetrického šifrování, je nutně šifrovací klíč součástí viru. (Nemusí to být úplně pravda, klíčem může být třeba i nějaký soubor, který není součástí viru. Ale je nutné, aby vir ke klíči měl přístup, tedy musí být stále dostupný.) A protože tímto klíčem

lze data i dešifrovat, nepředstavuje takový vir velikou hrozbu. Ihned po tom, co se ví, kde je klíč uložen, mohou být zašifrovaná data zachráněna.

Zajímavé to je u algoritmů, které používají veřejný a privátní klíč. Privátní klíč zná pouze autor viru a samotný vir šifruje pomocí veřejného klíče. Aby bylo možné data dešifrovat, je nutné mít privátní klíč. Vir ho však nevyžaduje a pokud napadený uživatel data nutně potřebuje, nemá moc možností. Buď doufat, že privátní klíč bude odtajněn, nebo přistoupit na dohodu s autorem viru.

O tom, že tato hrozba není pouze teoretická, svědčí výzkumné práce Adama Younga a Motiho Yunga, kteří v laboratorních podmínkách tento druh viru vytvořili. Jeho délka byla necelých 7 kB a podstatnou část kódu tvořila knihovna pro kódovací algoritmus RSA.

Citovaný kryptovirus zejména upozornil na paradoxní skutečnost, že vestavěná kryptografická podpora na úrovni operačního systému nejen přestává zvyšovat bezpečnost systému, ale naopak ji začíná snižovat v okamžiku, kdy takový systém není dostatečně odolný vůči virovým infiltracím. Obsahuje-li totiž hostitelský systém takovou podporu, pak tvůrce viru nemusí vůbec nic vědět o kryptografii, ale stačí mu pouze dané funkce volat.

6.5 Budoucnost malware

Nyní se vývoj ubírá dvěma směry:

1. První představuje jednoduchý malware, obvykle vytvořený pro jeden konkrétní operační systém a využívající jednu konkrétní bezpečnostní díru. Do této kategorie bych zařadil i červy posílané e-mailem nebo jako zprávy přes ICQ. Od uživatele vyžadují, aby je spustil, teprve díky tomu se šíří. Je až s podivem, jaký mají úspěch.

2. Druhým směrem pak bude komplexní a složitý malware. Nyní nemyslím jenom první variantu doplněnou o možnost šířit se na více systémů, případně zneužívat více bezpečnostních děr. Mám na mysli programy, které mezi sebou komunikují, aktualizují se a možná se dokonce specializují.

   Mezi základní principy, které bude nový malware využívat, bude patřit [3]:

   - Přenositelnost – platformová nezávislost a schopnost fungovat na různých operačních systémech

- Neviditelnost – používání stealth technik, ztěžování detekovatelnosti antiviry a samozřejmě i uživatelem
- Nezávislost – šíření automaticky bez zásahu uživatele, třeba používáním vestavěné databáze exploitů
- Učení – možnost učit se nové věci, aktualizovat kód, používat plug-iny, komunikovat a stahovat nové informace ze sítě od ostatních instancí
- Integrita – strukturu malware a jeho komunikační sítě musí být těžké vysledovat, upravit, rušit nebo zničit
- Polymorfismus – tělo musí být plně polymorfní, každá instance musí vypadat jinak
- Použitelnost – malware musí být schopen plnit zadané příkazy – například dostane instrukce, které provede, a pak se smaže ze všech systémů

Je dokonce možné, že budoucí malware bude vypadat jako „chobotnice“, bude se skládat z více programů, kde každý bude běžet na jiném počítači. Jedna část bude sloužit jako hlavní a ostatní budou fungovat jako její končetiny. Jestliže bude řídící část odstraněna, budou ostatní odříznuty, čímž se zvýší zranitelnost. Pokud ale bude každá část i součástí komunikační sítě, bude možné je připojit k jiné řídící jednotce nebo požádat jinou takovou skupinu o její vytvoření. Tímto způsobem se pak minimalizuje možnost odhalení, protože na napadených počítačích nebude kompletní kód a bude obtížnější ho detekovat.

Dále se nabízí šance na vytvoření komunikačního protokolu pro všechen malware od různých autorů. Pokud by mohly mezi sebou komunikovat, případně si i pomáhat, nejen se šířením ale i s poskytováním informací, jako jsou e-mailové adresy případně IP adresy dalších cílů. Dokonce by si mohly mezi sebou vyměňovat plug-iny nebo části kódů, čímž by napodobily živé organismy. Díky těmto výměnám by vznikaly nové zatím nedetekovatelné varianty, bohužel ale také některé méně životaschopné.

Použití komunikace po síti s sebou přináší problém zabezpečení komunikace. Nestačí pouze přenášená data šifrovat, protože pokud by se do takové sítě připojil nový útočník (nemusí to být nutně někdo z antivirové společnosti, může to být nějaký další autor malware), tak bude jednoduše šifrovat a dešifrovat také a nebude mu dělat problém vydávat se za někoho jiného. Případně bude falšovat informace, plug-iny a seznamy instrukcí, které mají jednotlivé instance malwaru provádět. Zabezpečit komunikaci proti tomuto zneužití je lehké, stačí používat elektronický podpis. Díky použití privátního klíče může odesílatel zaručit integritu zprávy (nikdo ji během přenosu nezměnil) a její nepopíratelnost (že je zpráva od něho, jelikož nikdo jiný privátní klíč nezná). Pro ještě větší zvýšení

bezpečnosti se může přidat i časové razítko (zpráva pak nepůjde použít později, pokud by byla odposlechnuta a poslána znovu).

Neuronové sítě patří mezi technologie, které se dnes začínají používat ve velkém. Neuronové sítě jsou napodobeniny mozku, kde je podstatné, že se umí učit a vyvozovat závěry ze zkušeností. Této schopnosti adaptovat se na prostředí tímto způsobem se říká inteligence. Třeba se časem setkáme s malwarem, který bude sám objevovat nové bezpečnostní díry, sám sebe upravovat a přidávat si nové schopnosti. [5]

7 Závěr

V mé bakalářské práci jsem se snažil zhodnotit vývoj malware. Abych však mohl vývoj a budoucnost malware zanalyzovat, musel jsem nejprve představit jeho jednotlivé typy a popsat jeho historii.

K vytvoření přehledu druhů malware jsem použil již zmíněnou klasifikaci podle způsobu šíření. Mnou uvedené schéma je neoficiálně přijaký standard, jelikož podobným způsobem malware pojmenovává většina antivirových společností.

Ve třetí kapitole jsem popsal časovou osu, která není zdaleka kompletní, o což jsem se ani nesnažil. Mým cílem bylo ukázat, jak se postupně objevovaly novější programy, které využívaly starší algoritmy a přidávaly k nim nové funkce.

Rád bych zde zmínil pro mě nejzajímavější malware, které jsem si vybral k detailnějšímu popisu. Snažil jsem se, aby zapadaly do tématu mé práce.
První – OneHalf – je příklad kryptoviru. Jeho autor šifruje data na disku a za jejich záchranu vyžaduje zaplatit. Pokud by místo symetrického šifrování používal šifrování veřejným klíčem, byla by obnova dat bez pomoci autora viru prakticky nemožná.
Druhým je červ Hybris. Když jsem o něm poprvé uslyšel, ihned mne fascinoval. Používá plug-iny, kterými si může rozšířit své funkce a zároveň se aktualizovat. Takto komplexní malware mohl být vytvořen jenom spoluprácí více autorů. Jedinému člověku by práce na něm trvala velmi dlouho. Ukazuje se, že takováto spolupráce není nemožná a nemusí nutně vést k jednomu složitému programu, jak je vidět v podkapitole 6.2. Někdy je výhodnější kvantita nad kvalitou. Jeho autoři přidávají nové funkce a tím mírně pozměňují původní verze malware, což může vést k novým, nedetekovatelným a životaschopnějším kusům. Zároveň vlastní změna není tak časově náročná jako přepsání celého kódu.
Červ Slapper jako jediný, zde uvedený, nenapadá systémy Microsoft Windows, ale Linux. Využívá bezpečnostní chyby v knihovně, která sama má bezpečnost zvyšovat. Ale hlavní důvod, proč ho zde uvádím, představuje jeho schopnost vytvořit síť jím napadených počítačů. Tato síť se nazývá botnet a zastupuje jeden z hlavních trendů vývoje malware.
Jako poslední vzorek zmíním vir Zmist, který použitím mnoha různých technologií obrany dlouho unikal odhalení antiviry. Mimo, v té době běžného, polymorfismu a metamorfismu používá také integraci kódu, kterou objevil jako první malware. Další z jeho ne nepodstatných vlastností bylo, že se náhodně rozhodoval, jestli danou činnost provede. To představuje velmi silnou obranu proti

emulaci kódu antiviry, neboť tím, že se spustí jenom někdy, se může vyhnout odhalení. Samotné generování náhodných čísel (často se používá algoritmus Mersenne twister, který dosahuje velmi dobrých výsledků a zároveň je rychlý a lehce implementovatelný) se využívá i u mnoha jiných činností. Například u metamorfismu, kde se mají povyměňovat jednotlivé části kódu, je vhodné, aby vzniklo co nejvíce rovnoměrně rozložených kombinací.

Sociální inženýrství a naše každodenní zkušenosti ukazují, že se i přes vynikající zabezpečení počítače může dostat kterýkoli malware. Stačí jediné neuvážené kliknutí a systém je vystaven právě spuštěnému programu. Je také pravděpodobné, že si sám stáhne mnoho jiných ještě nebezpečnějších. Navíc sníží bezpečnost vytvořením zadních vrátek, které umožní útočníkovi provádět s tímto počítačem, co si bude přát. Závěr této kapitoly pak popisuje novou hrozbu – spolupracující malware, který využívá sociotechniku naplno. Zatím jsem se o něm v žádné litaratuře nedočetl. Je pouze otázkou času, kdy vznikne. Ve skutečnosti k jeho existenci ale zatím není důvod. Člověk se až diví, jak někdo může přijmout e-mail z neznámé adresy, uložit dvě přílohy, obrázek a ZIP archiv. Z obrázku zjistit heslo, rozbalit a dešifrovat archiv a takto vzniklý soubor spustit. Navíc když celý text e-mailu s návodem, jak to udělat, je v angličtině.

V poslední kapitole pak popisuji aktuální hrozby, jako jsou například boty, které se spojují do větších botnetů. Takto obrovské botnety jsou vytvářeny kvůli zisku. Ten pramení především z rozesílání spamu a phishingu. Ačkoliv by se mohlo zdát, že tyto zprávy rozesílá několik tisíc lidí, ve skutečnosti většina pochází od maximálně desítky lidí. Od majitelů velikých botnetů, kteří ovládané počítače používají právě k tomuto rozesílání. Schopnost ovládat tisíce počítačů rozmístěných různě po světě a připojených k internetu poskytuje neuvěřitelné možnosti.

Škodlivé programy existují již dlouhá léta a jistě se budou i nadále vyvíjet. Pokud s nimi nechceme ve vzájemném souboji prohrát, musíme se více zaměřit na informovanost veřejnosti a větší zabezpečování počítačů. Dle mého názoru tuto válku sice vyhrát nemůžeme, ale můžeme se pokusit s malwarem držet krok. Věřím, že moje práce bude užitečná a v tomto boji nápomocná.

Literatura

[1] Szor, P. *The Art of Computer — Virus Research and Defense.* Symantec, 2005.

[2] Wikipedia, http://wikipedia.org/.

[3] Zalewski, M. Projekt Samhain. http://lcamtuf.coredump.cx/worm.txt.

[4] Krčmář, P. http://www.root.cz/clanky/boinc-distribuovane-vypocty-doma/.

[5] Myslík, P. Referát z předmětu speciální architektury 1996/1997, http://aldebaran.feld.cvut.cz/~xmyslik/www/neural.html.

[6] Mitnick, Kevin D., Simon, William L. Umění klamu. HELION S.A., 2003.

[7] Bond, M., Danezis, G. A pact with the Devil. Technical report UCAM-CL-TR-666, University of Cambridge, Computer Laboratory, June 2006.

[8] Jalůvka, J. *Moderní počítačové viry.* Computer Press, 2000.

[9] Singh, S. *Kniha kódů a šifer.* Nakladatelství Dokořán a Argo, 2003.

[10] Hák, I. Moderní počítačové viry. http://viry.cz/viry.cz/kniha/kniha.pdf.

[11] Šimek, R. Sociotechnika (sociální inženýrství). http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xsimek3sociotechnika.htm.

[12] Nykodýmová, H. Botnety: nová internetová hrozba. http://www.lupa.cz/clanky/botnety-internetova-hrozba/.

[13] Hammond, S. Busting the botnet-herders. http://www.cw.com.hk/computerworldhk/article/articleDetail.jsp?id=190067.

[14] Gostev, A. Malware Evolution: January - March 2007. http://www.viruslist.com/en/analysis?pubid=204791938

Seznam příloh

Příloha 1. DVD s elektronickou verzí této práce