Linux Viruses and Such

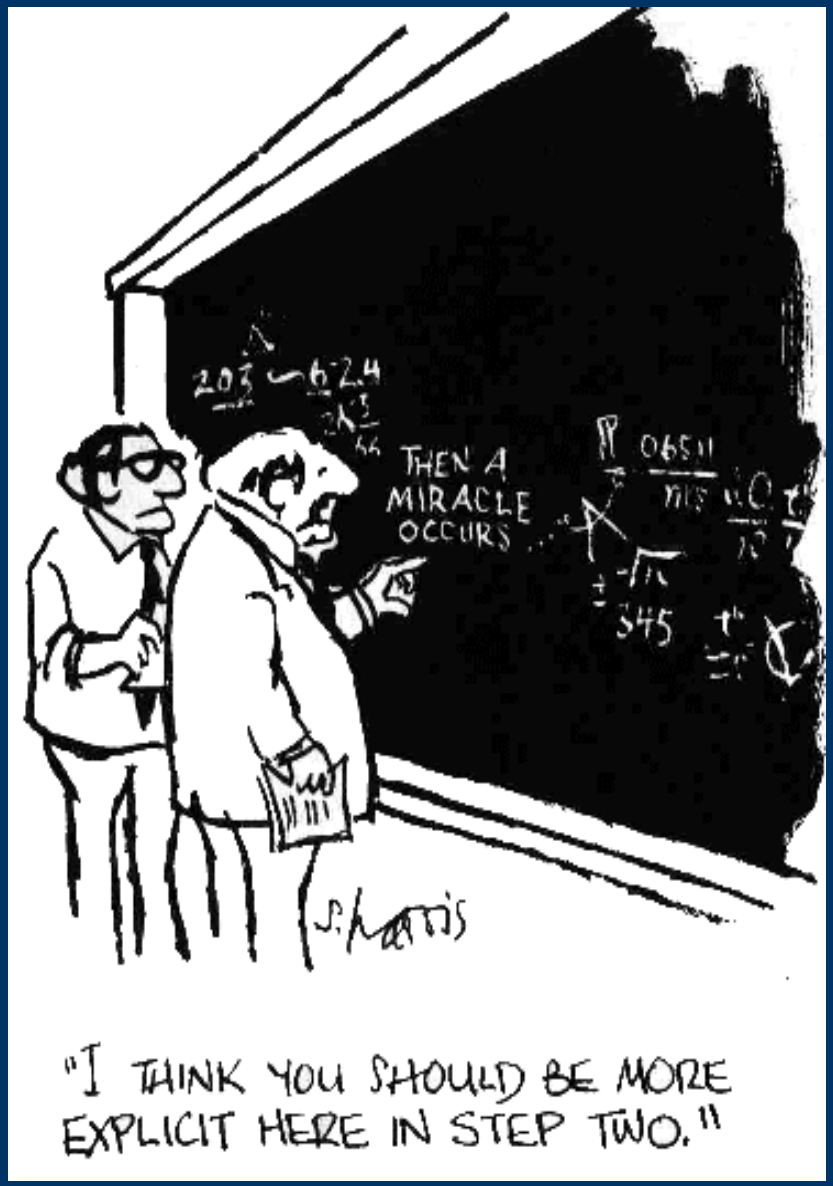
EBLUG, 2004-12-15

Challenges in writing this talk:

- Writing a Linux talk (avoiding giving Yet Another Windows Virus Lecture).
- Explaining any one thing first.
- Many antivirus and security people are not in the perspective business.
- Dealing with the gadget freaks among us.
- The Net's collective memory is timeline-impaired.

Note: This is not^W a security lecture in disguise.

Linux Viruses and Such



Most virus press coverage & analysis is like this cartoon.

Malware requires:

- authorship
- entry
- execution
- a vector

The first is easy on Linux, the second conceivable. The latter two tend to be non-trivial.

Linux Viruses and Such

Malware Types:

- Virus: A program that modifies other programs.
- Worm: Program that propagates across networks.
- Trojan: Program that purports to do one thing, but does another. ("Spyware" codebases are basically trojans.)
- Logic bomb: Program that "blows up" in some way. (Milder versions are called "bacteria".)

Linux Viruses and Such

Non-malware:

- Rootkit: Post-compromise toolkit to hide an attacker's presence after he enters *via other means entirely*.
- [D]DoS tool: [Distributed] Denial of Service against remote host.
- Backdoor: Post-compromise covert means of re-entry for an attacker.
- Documented but misunderstood: E.g., “sendmail -C”.

Linux Viruses and Such

Four thoughts to ponder:

- How do you know your system isn't compromised (virus or no virus)?
- Viruses are canaries. (Thank them for pointing you towards the real problem.) Measures to prevent or limit damage are same as for any other process.
- Code doesn't run itself.
- Security forensics is difficult; bad guesses are easy.

Linux Viruses and Such

Dept. of Reddish Herrings (errors & irrelevancies):

- “There are no Linux viruses.”
- “Linux isn't invulnerable!”
- “Unix is designed for security.” Dennis Ritchie, in “On the Security of Unix”: “The first fact to face is that UNIX was not developed with security, in any realistic sense, in mind; this fact alone guarantees a vast number of holes.”
- “All I have to do is get the user to do [foo].”
- “There's no protection against social engineering.”
- “Open source will protect us.”

Linux Viruses and Such

Dept. of Reddish Herrings (cont'd)

- “Viruses are difficult to write.”
- “Scripting support is dangerous.”
- “Someone could write Outlook for Linux.”
- “Data files can have executable code in them!”
- “Hordes of new Linux users spell doom.”
- “Security doesn't work, because users are lazy and do stupid things.”
- “You're safe if you use the root account carefully.”
- “You're safe if you patch all known bugs.”

Linux Viruses and Such

Dept. of Reddish Herrings (concluded)

- “You have to worry about cross-platform threats from {Win32emulators|Java|JavaScript|Office Suites|Perl|PostScript|Python|Tcl|TeX|elisp|Web browser runtimes}.”
- “All you have to do is post evil source code on Freshmeat.”
- “Linux is too insignificant to attack.” (Tell that to all the people running Apache httpd.)

Linux Viruses and Such

"Viruses are canaries" example:

Jan 25, 2003, 9:00 a.m., Saturday: Davis-Besse nuclear power plant, near Toledo, Ohio:
SQL Slammer worm for Windows enters and overwhelms the SCADA network. Davis-Besse's safety-monitoring system was down for five hours.

North American Electric Reliability Council reported that two other electric plants' (one hopes, non-nuclear) SCADA networks were also affected.

Hooray for the virus! (relatively speaking)

Linux Viruses and Such

'The Slammer worm entered the Davis-Besse plant through a circuitous route. It began by penetrating the unsecured network of an unnamed Davis-Besse contractor, then squirmed through a T1 line bridging that network and Davis-Besse's corporate network. The T1 line, investigators later found, was one of multiple ingresses into Davis-Besse's business network that completely bypassed the plant's firewall, which was programmed to block the port Slammer used to spread.'

'"This is in essence a backdoor from the Internet to the corporate internal network that was not monitored by Corporate personnel," reads the April NRC filing by FirstEnergy's Dale Wuokko. "[S]ome people in Corporate's Network Services department were aware of this T1 connection and some were not"'

Linux Viruses and Such

Malware Timeline:

- 1949: John von Neumann invented the concept of self-reproducing automata.
- 1969 onwards: Ken Thompson built into binaries of the original C compiler a backdoor for the UNIX login program, for his own convenience. No amount of source checking could find it, and it was still undetected when he revealed it in his 1983 Turing Award lecture.
- 1972: Novel “When HARLIE Was One” by David Gerrold had a fictional program called VIRUS, which was a war dialer that “injects itself into the new computer”.
- early 1970s: Bob Thomas at BB&N created the Creeper program, which traveled from computer to computer within the network, then the Reaper program to hunt it down and kill it.

Linux Viruses and Such

Malware Timeline (cont'd)

- mid-1970s: Motorola staffers discovered exploitable security hole in the Xerox CP-V timesharing system, but *couldn't get Xerox to fix it*. So, the Motorola guys unleashed daemon programs “Robin Hood” and “Friar Tuck”, which carried out flamboyant pranks and re-launched one another if the system operator tried to kill them:

```
!X id1
id1: Friar Tuck... I am under attack!  Pray save me!
id1: Off (aborted)

id2: Fear not, friend Robin!  I shall rout the Sheriff
     of Nottingham's men!
id1: Thank you, my good fellow!
```

Linux Viruses and Such

Malware Timeline (cont'd)

- 1975: Novel “Shockwave Rider” by John Brunner invented a fictional “tapeworm” program.
- 1981: Elk Cloner by Richard Skrenta. Simple floppy/OS infector for Apple II AppleDOS 3.3, and first non-fictional virus. Every 50th time it was run, an infected program displayed:

“Elk Cloner: The program with a personality
It will get on all your disks
It will infiltrate your chips
Yes it's Cloner!
It will stick to you like glue
It will modify ram too
Send in the Cloner!”

Linux Viruses and Such

Malware Timeline (cont'd)

- 1982: John Shoch and John Hupp of Xerox PARC wrote an experimental worm program running across 100 Alto computers over ethernet. It ran amok because of a coding error, and had to be shut down.
- Nov. 3, 1983: Fred Cohen wrote an experimental virus for a VAX 11/750 running 4BSD, to present to the Nov. 10, 1983 meeting of a weekly seminar on computer security, part of the work for his doctorate at USC. Leonard Adleman, his seminar advisor, applied the term “virus” to the work, published in 1984. Virus used trojan techniques to (hypothetically) spead it: It was embedded in new debugging utility “vd”, posted on the system bulletin board.

Linux Viruses and Such

Malware Timeline (cont'd)

- 1983: Film “War Games” introduced the public to the concept of a computer backdoor, and of war dialers.
- 1986: First MS-DOS virus, Brain, by brothers Basit and Amjad Alvi of Lahore, Pakistan. Infected floppy boot sectors.
- 1987: First file infector, Lehigh (infected command.com). Later, Suriv-02 was first .EXE infector (later reworked as "Jerusalem").
- 1987: First e-mail worm, Christmas Exec, for IBM VM/CMS mainframes, forced shutdown of several systems.
- 1987: U. of Wellington student unleashed Stoned, the first MS-DOS MBR virus. (Displayed “Your PC is now stoned!” and “Legalise Marijuana!”)

Linux Viruses and Such

Malware Timeline (cont'd)

- Nov, 2, 1988: Morris worm. Specific to VAXen and SUN3 boxes running BSD. Attempted fingerd buffer overflow, did password guessing, attempted to use sendmail DEBUG option.
- 1989: AIDS, by PC Cyborg Corporation of Panama: first trojan program.
- 1992: Michelangelo: Pioneer virus-industry press frenzy.
- 1994: Good Times hoax. (Cf. April 2000 “ILOVEYOU” worm.)
- 1995: Concept virus, by “Black Baron”. First macro virus.

Linux Viruses and Such

Malware Timeline (cont'd)

- Sept. 29, 1996: Bliss. First Linux virus. Third party posted it to Usenet groups comp.security.unix, alt.comp.virus and comp.os.linux.misc. ELF infector. *Most courteous virus ever:* Included "--bliss-uninfect-files-please" option. Can be compiled for any *ix variant.

Dishonest McAfee press release of Feb. 5, 1997 claimed to have "discovered" Bliss.

- Oct. 1996: Staog by "Quantum" of the VLAD virus group in Australia (published in VLAD Magazine issue #7). Often claimed in error to be the first Linux virus, because of a Feb. 5, 1997 Bliss "sighting" on linux-security mailing list that many mistakenly believed to be its first appearance. Staog probed for buffer overflow vulnerabilities in mount and tip, and a bug in suidperl, to try to gain root access.

Linux Viruses and Such

Malware Timeline (cont'd)

- 1998: Strange Brew. First Java virus.
- May 1998: Adm worm for Linux. Attacked hole in a BIND8 function that's never enabled, already fixed a month earlier.
- Jan. 21, 1999: TCP Wrappers v. 7.6 package trojaned at ftp.win.tue.nl, Eindhoven U. Detected within hours by Andrew Brown of Crossbar Security, who noticed it was unsigned. util-linux 2.9g also found trojaned at the same site, next day: John Stange noticed modification to login.c.
- 1999: Melissa. First Word/Outlook e-mail virus.
- Sept. 25, 2000: Red Hat Software, Inc. rolled out Red Hat Network auto-updating, default starting with Red Hat 7.0.

Linux Viruses and Such

Malware Timeline (cont'd)

- 2001-3: Eleven network worms aimed almost entirely at Red Hat. (Details at my site.) Attacked obsolete versions of wu-ftpd, BIND8, NFS, lpd, LPRng, OpenSSL, Samba, OpenSSH, qpopper, wu-imapd. All holes had been patched months or years earlier. Claim made (Ryan Russell) that two of these (1i0n, lpdw0rm) subverted “thousands” of RH boxen when est. # of Linux installations was 10-20 million. Pundit Laura DiDio asserts that 2002 Slapper/Cinik attack on OpenSSL w/Apache subverted 20,000. (Symantec says 3,500.)

Linux Viruses and Such

Malware Timeline (cont'd)

- May 6, 2002. Red Hat Software, Inc. started defaulting to enabling IP-filtering enabled, with its shipment of Red Hat 7.3.
- Mar-May 2003. monkey.org and irssi.org sites compromised, leading to backdooring of dsniff, fragrouter, fragroute, and Irssi source tarballs. Detected seven days later, GnuPG signing of source releases commenced. 2,000 downloads.
- Jan. 25, 2003. "SQL Slammer" worm, AKA Sapphire or Slapper, subverted an estimated quarter-million Windows boxes within about ten minutes. Saturated entire Internet in ½ hour.
- Nov. 5, 2003: Attempt failed to plant a backdoor into the Linux kernel via a subtle addition, disguised as a two-line enhancement to the sys_wait4 function's error-checking in file exit.c. Change planted at a CVS-checkout site (kernel.bkbits.net) housing copy of the BitKeeper tree was caught by an automated integrity check.

Linux Viruses and Such

Malware Timeline (concluded)

- Nov. 2003: Compromise of four Debian Project servers, a server participating in the Gentoo Project's rsync.gentoo.org cluster (but neither project's packages), and FSF's Savannah server, using stolen login credentials plus escalation to root using a recently discovered (Sept. 2004) bug in the v. 2.4.22 kernel's brk() system call: Andrew Morton had patched the bug without realising its security implications.

Both the Debian and Gentoo compromises were detected within hours because of (1) file-integrity checkers, and (2) alert sysadmins noting a suspicious pattern of kernel “oopses”.

Linux Viruses and Such

All known ELF infectors for Linux:

Abulia, Bliss, Cassini, Cron, Dido, Diesel, Dummy, Eriz, Gildo, Henky, Jac, Kagob, Kaot, Mandragore, Nel, Neox, Nuxbee, Obsidian.E (Obsid), OSF, Ovets, Pavid (Alfa.dr), Penguin, Quasi, RST = Remote Shell Trojan (Vit), Radix, RcrGood, Rike (Rike.1627), Satyr, Sickabs, Siilov, Silvio, Simile (Etap, MetaPHOR), Staog, Svat, Telf, Thebe, Winter (Lotek, LoTek), Winux (Lindose, PEElf, Pelf), Wozip, Xone, Ynit, and Zipworm (distinctive only in that it likes to infect ELF files in Zip archives).

Pretty much all functionally identical. Might as well be one virus. (A few include backdoor code, which please see.)

Linux Viruses and Such

Remedies / Lessons / Concluding Sermon:

- The "crunchy centre" approach: hardening executables.

libsafe, StackGuard, Stack Shield, FormatGuard, PointGuard, RaceGuard, PaX, ExecShield, RSX, Openwall, kNoX, grsecurity. All of these either modify gcc or interject a protective library.

Problem: Things tend to break.

Linux Viruses and Such

Remedies / Lessons / Concluding Sermon (cont'd):

- Sandboxing: Little used, except inside Java. Room for improvement to Linux systems, here -- e.g., in Internet “viewer” apps.

- Modular design, privilege-dropping, chrooting.

- Access Control schemes:
  Mandatory Access Control:
    SELinux, LIDS, RSBAC
  Discretionary Access Control:
    POSIX.1e draft (1003.1e/1003.2c) ACLs

Linux Viruses and Such

(Further sermon:) Surveillance of Self:

- Network IDSes / Vulnerability Scanners:
  Snort, nmap, Nessus

- Host-based IDSes / System Integrity Checkers:
  AIDE, Integrit, Samhain, Prelude-IDS, Tripwire.

- chkrootkit
  ...but note LKM stealth rootkits.

Linux Viruses and Such

(More sermon:) Competent system maintenance:

- *Do* use your packaging and maintenance regime. Avoid the tarball shtick absent a very compelling reason, and then realise you're assuming all the responsibilities of a packager. Also, install within /usr/local or /opt; avoid root authority if possible.
- *Don't* use obsolete and/or unmaintained software, e.g., BIND8.
- *Don't* use outright bad software, e.g., wu-ftpd.
- *Do* check your mime.types, mailcap, or equivalent.
- *Do* try to have some idea which code is more security-sensitive.
- *Do* know that you're responsible for whatever you run with whatever authority you wield in so doing.

Linux Viruses and Such

Remedies / Lessons / Concluding Sermon (almost done).

How about Running a Virus-Checker for Linux, Too?

Rick's Not-At-All-Cynical Guide to Distinguishing Viruses from Virus Checkers:

	Virus	Virus-Checker
Unauditable, binary-only code?*	yes	yes
That you're expected to run as root?	yes	yes
From someone who claims to be a good guy?	no	yes

*(Sure, I do know about ClamAV, AMaViS, etc.)

Linux Viruses and Such

Cluebat: Hey, Linux guy: You have much more real and serious worries than viruses. E.g.:

The authentication problem, passwords, and $FIRM's compromise as described in my file about “break-ins without remote vulnerability”.

Advantages of multifactor authentication. Maybe S/KEY or OPIE?

Linux Viruses and Such

We are protected by:

- Ground-up design for network and multiuser.
- Avoidance of system dependency on RPC calls.
- Modularity and generic, simple interfaces (scripting, pipes, etc.).
- Minimum privilege, no easy paths to escalation.
- System transparency.

(...in addition to just avoiding shooting at our feet.)

Linux Viruses and Such

In contrast to the Windows detection/removal treadmill, real-world security should concern all of these:

- prevention/avoidance
- detection
- damage reduction (what is at risk?)
- defense in depth (how can we avoid having all our eggs in one basket?)
- hardening
- identification
- recovery

Linux Viruses and Such

References:

- North American Electric Reliability Council report on the SQL Slammer worm http://www.esisac.com/publicdocs/SQL_Slammer_2003.pdf
- Davis-Besse nuclear plant infection. http://www.securityfocus.com/news/6767
- Amit Singh's “A Taste of Computer Security” http://www.kernelthread.com/publications/security/
- Assembler source code to Staog: http://www.lisoleg.net/lisoleg/security/LINUX%20Virus(very%20old).txt
- Ken Thompson's “Reflections on Trusting Trust”: http://www.acm.org/classics/sep95/

Linux Viruses and Such

References (cont'd):

- BK2CVS gateway's compromise discovered by Larry McVoy: http://www.ussg.iu.edu/hypermail/linux/kernel/0311.0/0621.html
- Fyodor's gentle tutorial on port-scanning: http://www.insecure.org/nmap/nmap_doc.html
- The Motorola/Xerox saga of “Friar Tuck” and “Robin Hood”, in Appendix A of the Jargon File: http://www.catb.org/~esr/jargon/html/meaning-of-hack.html
- A virus written in TeX by Keith Allen McMillan: ftp://ftp.cerias.purdue.edu/pub/doc/viruses/KeithMcMillan-PlatformIndependantVirus.ps
- Alexander Bartolich's ELF Virus Writing HOWTO. (Ego alert: Author cites Rick Moen's Linux virus pages as “inspiring”.) http://virus.enemy.org/virus-writing-HOWTO/_html/index.html

Linux Viruses and Such

- References (cont'd):

- Rick Moen's “Attacking Linux”, on how to study your systems' security by approaching/studying them the same way an attacker would: http://security.itworld.com/4352/LWD000829hacking/pfindex.html

- Robert Graham's analysis of the SQL Slammer worm: http://www.robertgraham.com/journal/030126-sqlslammer.html

- *Psst! Want to download a Linux virus?* http://www.digitaloffense.net/worms/ http://vx.netlux.org/

- Rick Moen's “Constructive Paranoia at the End of 2003”, drawing the lessons of the 2003 compromises of prominent Linux sites using the brk() kernel bug and stolen login credentials: http://linuxgazette.net/issue98/moen.html

Linux Viruses and Such

References (concluded):

- Wichert Akkerman's outstanding Debian.org Compromise 2003 pages: http://www.wiggy.net/debian/developer-securing/
- Rick Moen's piece on “break-ins without remote vulnerability”: http://linuxmafia.com/faq/Security/breakin-without-remote-vulnerability.html
- Nick Petreley's analytical article “Security Report: Windows vs Linux”: http://www.theregister.co.uk/security/security_report_windows_vs_linux/
- Details and analysis of the eleven worms, almost entirely in 2001-2, directed at unmaintained RH 6.2 and prior:
- http://linuxmafia.com/~rick/faq/index.php?page=virus#virus5
- Slides for this talk: http://linuxmafia.com/presentations/