VIRUSES , INFECTIOUS CODE, ATTACKS ON UNIX and LINUX SYSTEMS

PRESENTED BY BARRY SCHUETZ
Linux Supporters Group Adelaide
September 2nd 2009

Some Early Infections. Were found on used floppy disks (5.25”,3.25”) before the use of the Internet. These were mainly used to exchange files and other information between users.

Infections were also found on BBS Bulletin Board-Driven software, then CDs and tape drives.

The Early Internet. Arpanet was also where infections lurked, along with email attachments plus….

Other Removable Media, eg: CDs, DVDs, software distributed on CDs, and DVDs, portable HDDs, Flash-Drives, etc.

NB: The Gammia Infection propagates, via removable media, especially Flash-Drives.

(5)

Infections and Malicious code Unix-Linux systems that use WINE are particularly vulnerable. WINE is an open-source compatibility package that allows certain Unix-Linux platforms to run another form of software. They are vulnerable because they can make a system susceptible to both Unix-Linux and another form of software, Worms, Trojans.
(3)

A VIRUS. Is a program that infects or destroys other programs, usually without your permission.

A WORM. Is a replicating piece of code that operates without your permission, though bugs in your computer program may generate self-replicating code without your permission. The difference is that bugs are unintentional, and viruses are intentional.

A TROJAN. Hides the infection for the purpose of causing digital damage. In a Unix-Linux environment a Trojan can be given the extension of a legitimate program, eg: .tar or .txt, but many remove an entire file upon execution.

MALWARE. In broad terms malware is any type of software created to cause specific damage to a computer system or to circumvent the computer security. (all those mentioned in this paper)
(all 3)

VECTORS. Are mechanisms that spread malicious code infections.

NON RESIDENT VIRUSES. These viruses consist of a "finder-module" and a "replication module". The finder-module is responsible for finding new files to infect for each new executable file; when the finder-module encounters one it calls in the replication module to infect the files.
(5)

RESIDENT VIRUSES Contain a replication-module similar to the one used in the above. This module however is not called by the finder-module. This virus loads the replication module into memory so that it can remain active, or be activated, even after the program ends. Resident viruses are sometimes sub-divided into categories: of 'fast infectors' and 'slow infectors'.

FAST INFECTOR. Can pose a problem when using anti-virus software, since a virus scanner will access every potential host file on a PC when scanning. However if the scanner fails to notice that a virus is present in the memory, then the virus can piggy-back on the virus scanner and infect all those scanned files.

SLOW INFECTOR. These are designed to infect the hosts infrequently, and they are designed to avoid detection by limiting their actions they are less likely to slowdown a Pc noticeably. They will at most infrequently trigger anti-virus software that detects suspicious behaviour by programs. The slow infector approach, however does not seem very successful.
(5)

VIRUS SIGNATURES. Most modern anti-virus programs try to find a virus pattern inside ordinary programs by scanning them for so-called virus signatures. A signature is a characteristic byte-pattern that is part of a certain virus, or family of viruses. If a scanner finds such a pattern in a file it notifies the user that the file is infected.

(5)

ENCRYPTION with a VARIABLE KEY. A more advanced method is the use of simple encryption to encipher the virus. In this case the virus consists of a small decrypting module and a encrypted copy of the virus

code. If the virus is encrypted with a different key for each infection file, the only part of the virus that remains constant is the decrypting module.
(5)

POLYMORPHIC VIRUSES Use transformation engines that alter the digital signature of the virus each time it runs. This makes it much more difficult to detect since anti-virus engines cant simply search for a specific code string.
(1)

The SIMILE D Virus. May be particularity elusive because it isn't merely polymorphic; it combines metamorphic behaviour with its polymorphic code. This virus has over 14,000 lines of assembly code, 90% of which is part of the metamorphic engine. (1)

METAMORPHIC VIRUSES. Are even more slippery, changing all their code, and they don't contain a descriptor. They are difficult to find and pose a major threat to enterprise networks.

CROSS PLATFORM VIRUSES. Simile D (a.k.a. E trap D) attacks the same vulnerabilities in both Linux and other operating systems. With another system it attacks PE files and ELF, also it attacks Linux ELFs.
(1)

UNIX and LINUX INFECTIONS. The first major infection was launched in 1988: the "Morris Worm", also known as "network". It was written by a student at Cornell Uni., Robert Tappan Morris, and launched from MIT. Morris is now an associate professor at MIT.
This worm worked by exploiting known vulnerabilities in Unix: in sendmail, finger, and also weak passwords. The main body of the worm could infect machines running BSD4 and Sun systems 3.
The defence against this was inspired by Michael Robins Mantra, it was called "Randomization". About 6,000 machines were infected by the "Morris Worm".

BLISS.1996. This Trojan worm attempted to attach itself to Linux executables, for which the user has permissions. This virus was written to prove that Linux could be vulnerable. However Bliss doesn't have the ability to propagate with any efficiency, due to the complex structure of the user privilege system, though it is one of the only Linux viruses to be seen "in the wild"*. Bliss never reached wide popularity.
*(In the wild: That is outside the single computer, or Lab where it was created.)
(6)

ADM 1998. (admworm,) Vulnerable: bind 8 buffer overflow. Prior to 8.12 in the reverse query function, "fake-query yes, which is always disabled by default. The hole in question had been fixed for only a month, which might have made it a plausible threat, except that "fake-query" is pretty much always disabled.

(10)
LION 2001. Vulnerable: bind 8 to 8.2.3, via the TSIG* exploit of Jan 29 2001. Note: bind 9, initial release 15.9.2000; bind 9.1.0 release 17.1.200. *(transaction signatures securing DNS)

IPDWORM 2001. Vulnerable: (kork. abditive) Berkley lpd printing package, via input validation bug. Fixed in lpd Oct 2000 release.
Both Berkeley lpd and bind 8 were notoriously buggy network daemons, and neither was necessary or recommended unless you were running particular types of server machine. If you ran them anyhow, pretty much everyone advised you to always stay absolutely current on security fixes. Fortunately, the above worms were no threat, and the holes they attack were already fixed two and six months earlier.

SLAPPER 2002. (cinik, unlock, bugtraq.C) Apache/mod_ssi worm Vulnerable; A very specific and rare combination of Apache httpd with open SSL 0.9.6/0.9.7 beta 1, or earlier, via an open ssl buffer over flow. (Fixed 2.7.2002.) This worm attacks only e-commerce and other ssl-enabled web sites with particular obsolete versions of open-ssl and apache httpd, configured in a particular way, and the exotic hole it attacks had already been fixed for two months. (all 10)

SSHD22.2001 Vulnerable: Open-ssh exploit effective prior to V.2.3.0 old versions were patched 27.2.01; 2.3.0 released Nov 2000. People already had this hole patched for either eleven or eight months, depending on whether they were willing to jump to V.2.3.0 or not.
(10)

SORSO.2003. Vulnerable: Samba prior to V.2.0.10/2.2.8a, via buffer overflow. Those fixed versions were released 7.4.2003.
This is the only Linux worm to date targeting Samba server role packages obsolete versions, possibly because even reckless server Admins tend to know another O/S system file print sharing isn't safe to make accessible to the global internet. The attack holes already have been fixed.

LUPPER.2005.(lupii, plupii, marc) Vulnerable: PHP xmlrpc messaging library V.1.1.1, via url input validation bug enabling execution of arbitrary php. Fixed 8.8.05

JINGLE BELLS. 2003.(jbellz) Vulnerable: The proprietary mpg123 music-playing apps. buggy non-production pre-0.59s beta, but not prior or subsequent production versions, via a buffer overflow induced by trojan

(specially malformed) mp3 files played using it. Binary code in the MP3 frame header invokes a shell and recursively deletes the user's home directory. Fixed same day - even prior pre.0.58r beta was immune - but didn't meet quality standards for inclusion in any Linux Distro.

OTHERS sendmail (mail server) oz and squirrel-mail 07. Some of these viruses may effect FreeBSD or Solaris.
(all 10)

Quotes from Rick Moen
"There are real threats to Linux Security, if you spend time looking for "Linux Viruses"- which by and large can come at your system only if you get behind them and 'push' - you might miss the real threats. Do something useful like studying your security profile and other measures.
"Yes some virus author could in principle, some day in the very worst-case scenario- if he/she were able to find a remotely exploitable Linux Kernel network-code flaw, unknown to everyone else. They could unleash a devastating and rapid automated surprise attack that clobbers (compromises) within one hour a large percentage of, say, worldwide internet connected i386 Linux Servers TCP/IP stacks, and thus gain "root" Control. This would force all afflicted systems to be offline for a day to await the necessary patch and be rebuilt. That would be very annoying but would hardly be unrecoverable. Moreover I'll (Rick) give very long odds against this, or less-central failures happening, and lower ones for the same threats against practically every other OS.

Why? Some of the reasons were articulated nicely in (separate) analyses by Nick Petreley, Eric Raymond, and Karsten M .Self.
(10)

- Linux System was designed for multi-user and networked operation from the ground up.
- The system was designed to distrust and not rely (in the general case) on remote procedure calls(RPCs) especially between hosts.
- The system is profoundly modular.
  Kernel updates.

For these points see the above mentioned web sites for more detailed and comprehensive write ups. (10)

DAVID F.SKOLL. of 'Roaring Penguin Software', has written a response to claims by some Anti-Virus Software Company Executives. About Linux…
….
"Linux will be a target because it's use is becoming more widespread" said Raimond Genes, European President for Anti-Virus at Micro Trend.

Jack Clarke, European Product Manager at McAfee said….."In fact it's probably easier to write a virus for Linux because it's all open-source and

the code is available. So we will be seeing more Linux as the OS becomes more common and popular."

D.F.S. said "I will be charitable and call these statements "Myths" or "Misperceptions" rather than other nastier, but perhaps more accurate, terms. Apache web server is far more widely used than IE, but has suffered far fewer security problems.
The U S National Security Agency provides a security enhanced Linux Distro which contains advanced security features beyond anything found in other systems.
I have given acknowledgement to Skoll's paper in my Reference Notes.
(2)

ATTACKERS. Use professional software to create, distribute and administer botnets, trojans, and viruses. These well trained and very organised community of intrusion specialists distribute user-friendly software to aspiring beginners.
It is impossible to maintain IT security without an understanding of the tools used by profession intruders.
The NVIDIA GE- force graphics card is used because it is especially suited for cracking passwords.
Companies from Russia and Eastern Europe, especially the older Soviet States, provide much of the sample hacker software; sys admins are not even aware of the tools used by intruders.
Every attacker, no matter how skilled they are, leaves tracks, that's where denying attackers a shell without admin. privileges could make an attack more difficult.
(8)

Vulnerability Scanner. Is a tool used to quickly check computers on a network for known weakness. Hackers also commonly use port scanners. These check to see which ports on a specified computer are "open" or available to access the computer, and sometimes will detect what program or service is listening on that port, and its version number.
(9)

Packet Sniffer. Is an application that captures data packets, which can be used to capture passwords and other data in transit over the network.
(9)

Script Kiddie. Is a non-expert who breaks into a computer system by using pre-packaged automated tools written by others. These are the outcasts of the hacker community. Also referred to as a Skiddiot.
(9)

SOCIAL ENGINEERING Is the art of getting persons to reveal sensitive information about a system. This is usually done by impersonating someone, or by convincing people to believe you have permissions to obtain such information.
An attack under Linux would go something like this: “save this file”, open up a shell, enable execute permissions on file, by typing…(chmod a +x filename) , and then run it by typing (./ filename)
(2)

ARP-SPOOFING. This puts an attacker in a position to sniff and thus manipulate local traffic. The so called ‘man-in-the-middle attack’ is a form of eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. These attacks are easy to perform, even with little knowledge of networking. (11)

Briefly How ARP Works. Address resolution protocol was to provide functionality. ARP maps IP addresses to a MAC address. Eg: if a client A needs to sent a packet to server B. they would need the MAC address of server B. eg: 192.168.11.8. The cache contains tables with IP address, and corresponding MAC addresses. (the table can hold static entries, eg those learnt from ARProtocol, dynamic entries are often valid for a short time only).
(11)

Internal Attackers. Curiosity, revenge, industrial espionage are all reasons why insiders attack systems on their own network. Sys-admins have a hard time preventing these internal attacks, because protecting the internal network is a lot more difficult than protecting against external attacks. (11)

ARP Packet. Is transmitted as the payload of the Ethernet frame.

MAC Spoofing. Is useful for attackers who want to protect their identity.

ARP Watch. An open source tool for Unix that monitors unusual ARP activities.

ARP Guard. Works within the framework of a sensor, which monitors ARP info, and can be used on small to large networks.

Hackers Categories. Use White Hat. Grey Hat, Black Hat: a spectrum of different categories. A black hat hacker is some one who subverts computer security, without authorization, or uses technology (usually a computer or the internet) for vandalism and malicious destruction.
(9)

Spoofing Attack. Involves one program, system, or web site successfully masquerading as another by falsifying data and thereby being treated as a trusted system by a user or other program. The purpose of this is usually to fool programs systems, or users into revealing confidential info, eg: passwords and usernames. (11)

Key Loggers. Is a tool designed to record every keystroke on an infected machine for later retrieval. Its purpose is usually to allow the user of this tool to gain access to confidential info. typed on the affected machine, such as passwords and other private data. They often use Virus, Trojans, and root- kit- like methods to remain active and hidden.
(9)

Clickjackers. A technique that allows hackers to display a 'fake web page' and overlay it with a legitimate site in a transparent layer, thereby fooling visitors into taking actions they didn't intend.

REFERENCES USED IN THIS PAPER.

1. http://www.zdnetasia.com/insight/software/o,39044822,39065520,00.htm
2. http://www.desktoplinux.com/articles/AT5785842995.html
3. http://www.zednet.com/indight/soa/linux-unix-viruses-demand-special-attention/0
4. http://www.en.wikipedia.org/wiki/list-of-linux-computer-viruses
5. http://www.en.wikipedia.org/wiki/computer-viruses
6. http://www.spamlaws.com/linux-viruses.html
7. http://www.linuxsecurity.com/quick /reference/guide
8. http://www.linux-magazine.com/W3/issue/102/092-093_hackers.pdf
9. http://www.en.wikipedia.org/wiki/hacker-(computer-security)
10. http://www.linuxmafia.com/~Rick/faq/index.PHP?page=virus
11. http://www.demuth.biz/veroeffentlichungen//traffic%20tricks%20-%arp%20spoofing%20and%20poisoning.pdf .

FURTHER READING.

http://www.zednet.com.au/insight/soa/top-10-linux-unix-vulnerabilities/0,139023731

http://www.vulnet.com/vnet/news/2115032/bug-watch-linux-safe-attack

http://www.cyber.com/detailsPHP?id22&sections=detailpapers

http://www.virus.bartalich.at/virus-writting-how-to/-html/intro.html

http://www.zednet.com.au/news/security/soa/slapper-worm-gains-strength-in-numbers

http://www.kernelthread.com/publiccations/security/vunix.html