Special Topics in Security
ECE 5698

Engin Kirda
ek@ccs.neu.edu

Northeastern University

Admin News

- Challenge 2 will go online right after class
  - You need to solve 2 out of 4 programs to get full points
  - One of them very similar to what I will show in class today

SUID Programs

- Each process has *real* and *effective* user / group ID
  - usually identical
  - real IDs
    - determined by current user
    - `login`, `su`
  - effective IDs
    - determine the “rights” of a process
    - system calls (e.g., `setuid()`)
  - `suid` / `sgid` bits
    - to start process with effective ID different from real ID
    - attractive target for attacker

- You cannot use SUID shell scripts anymore

DEMO, SUID program…

Shell Tricks Demo

Resource Limits

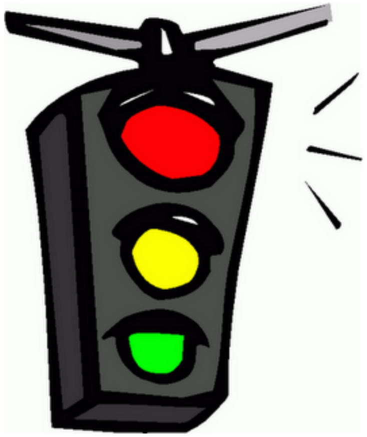
- File system limits
  - restrict number of storage blocks and number of inodes
  - hard limit
    - can never be exceeded (operation fails)
  - soft limit
    - can be exceeded temporarily
  - can be defined per mount-point
  - defend against resource exhaustion (denial of service)

- Process resource limits
  - number of child processes, open file descriptors

DEMO, quotas…

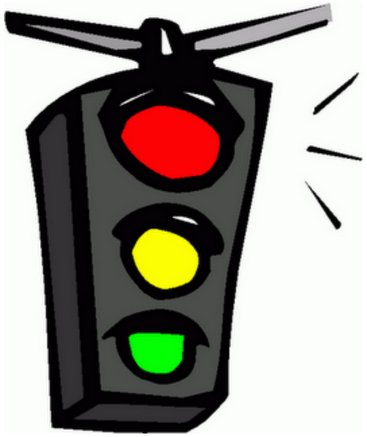
Signals

- Signal
  - simple form of interrupt
  - asynchronous notification
  - can happen anywhere for process in user space
  - used to deliver segmentation faults, reload commands, …
  - `kill` command



- Signal handling
  - process can install signal handlers
  - when no handler is present, default behavior is used
    - ignore or kill process
  - possible to catch all signals except SIGKILL (-9)

Signals



- Security issues
  - code has to be be re-entrant
    - atomic modifications
    - no global data structures
  - race conditions
  - unsafe library calls, system calls
  - examples
    - wu-ftpd 2001, sendmail 2001 + 2006, stunnel 2003, ssh 2006

- Secure signals
  - write handler as simple as possible
  - block signals in handler

Shared Libraries

- Library
  - collection of object files
  - included into (linked) program as needed
  - code reuse

- Shared library
  - multiple processes share a single library copy
  - save disk space (program size is reduced)
  - save memory space (only a single copy in memory)
  - used by virtually all Unix applications (at least libc.so)
  - check binaries with `ldd`

Shared Libraries

- Static shared library
  - address binding at link-time
  - not very flexible when library changes
  - code is fast
  - depends on kernel version

- Dynamic shared library
  - address binding at load-time
  - uses procedure linkage table (PLT) and global offset table (GOT)
  - code is slower (indirection)
  - loading is slow (binding has to be done at run-time)
  - classic `.so` or `.dll` libraries

- PLT and GOT entries are very popular attack targets
  - more when discussing buffer overflows

Shared Libraries

- Management
  - stored in special directories (listed in `/etc/ld.so.conf`)
  - manage cache with `ldconfig`

- Preload
  - override (substitute) with other version
  - use `/etc/ld.so.preload`
  - can also use environment variables for override
  - possible security hazard
  - now disabled for SUID programs (old Solaris vulnerability)

Unix / Linux Best Practices

- If you are maintaining a UNIX-based system…
  - – Turn off unused services
    - Services that are not enabled cannot be attacked
    - Services may be vulnerable (e.g., the printer example)
    - You might want to check *inetd (/etc/inetd.conf), /etc/init.d*

```
#pop stream tcp nowait root /etc/uva/tcp_wrapper/tcpd /usr/local/etc/popper
popper  #imap stream tcp nowait root /etc/uva/tcp_wrapper/tcpd /usr/local/etc/imapd4
imapd
```

- If you use xinetd, check /etc/xinetd

```
service finger{
socket_type = stream    wait = no    user = nobody    server =
/usr/sbin/in.fingerd    disable = yes
}
```

Unix / Linux Best Practices

- Install IP filter or firewall rules…
  - Even back in 2002, some UNIX systems were open (!)
  - *ipchains* is available with Linux 2.2, as of 2.4, *iptables*
  - AIX and IRIX have similar filtering capabilities
  - In Solaris, IP filtering is not part of the OS until version 8
    - You can buy it ;-) – Eeehmmm… maybe not a good idea
- Install *tcpd*
  - tcpd is a wrapper daemon for tcp-based services
  - With a configuration file, one has fine-grained control over accesses

Unix / Linux Best Practices

- Install *sshd*
  - ssh is stable and secure.
    - ssh has a good reputation
    - Nevertheless, there have been problems in the past (so patch your system)
  - Ideally, passwords should not be typed (on keyboard) and remote root access should be disabled
  - ssh in combination with IP-based restrictions and public-key configurations is a good idea
- Try not to use “web application frameworks” and the like
  - Some of them are riddled with holes (e.g., Wordpress)
  - E.g., Dan Kaminsky, Kevin Mitnick, sites, etc.

Unix / Linux Best Practices

- When you leave your desk…
  - You can use a screensaver with password protection
  - Unix systems often allow you to “lock” the screen
  - On MacOS, it might be a good idea to activate (multi-user login)
  - These things might sound trivial, but industrial espionage is an issue and unlocked computers are sometimes used to gain access
    - E.g., disguised cleaning lady
    - Companies often have checks and guidelines for desk management

One Advantage of Linux Today

- No matter how popular it has become, it still has a small number of users (compared to Windows)
  - If you use Linux today, you have a very very small risk of getting infected by a drive-by download
  - Malware for Linux exists, but most attacks are server-side and do not target end-users
  - A Linux VM is an ideal tool for accessing online banking
  - UNIX machines do not run as root by default
  - Using MacOS is less safe (mainly because it is more popular) – MacOS malware has appeared

The Most Common UNIX Attack

- Brute force attacks against services such as ssh, ftp, telnet
  - If you check server logs, frequently, you see repeated attempts for random user names (e.g., admin, root, etc.)
  - These are often bots who try brute force attacks against Internet hosts
  - A simple defense technique: Run your SSH server on a different port (e.g., 800)
    - The downside: Firewalls might be problematic

Random Number Bug in Debian

- On May 13th, 2008 the Debian project announced that Luciano Bello found an interesting bug
  - The random number generation was flawed in *md_rand.c*
  - The following lines were removed from code:

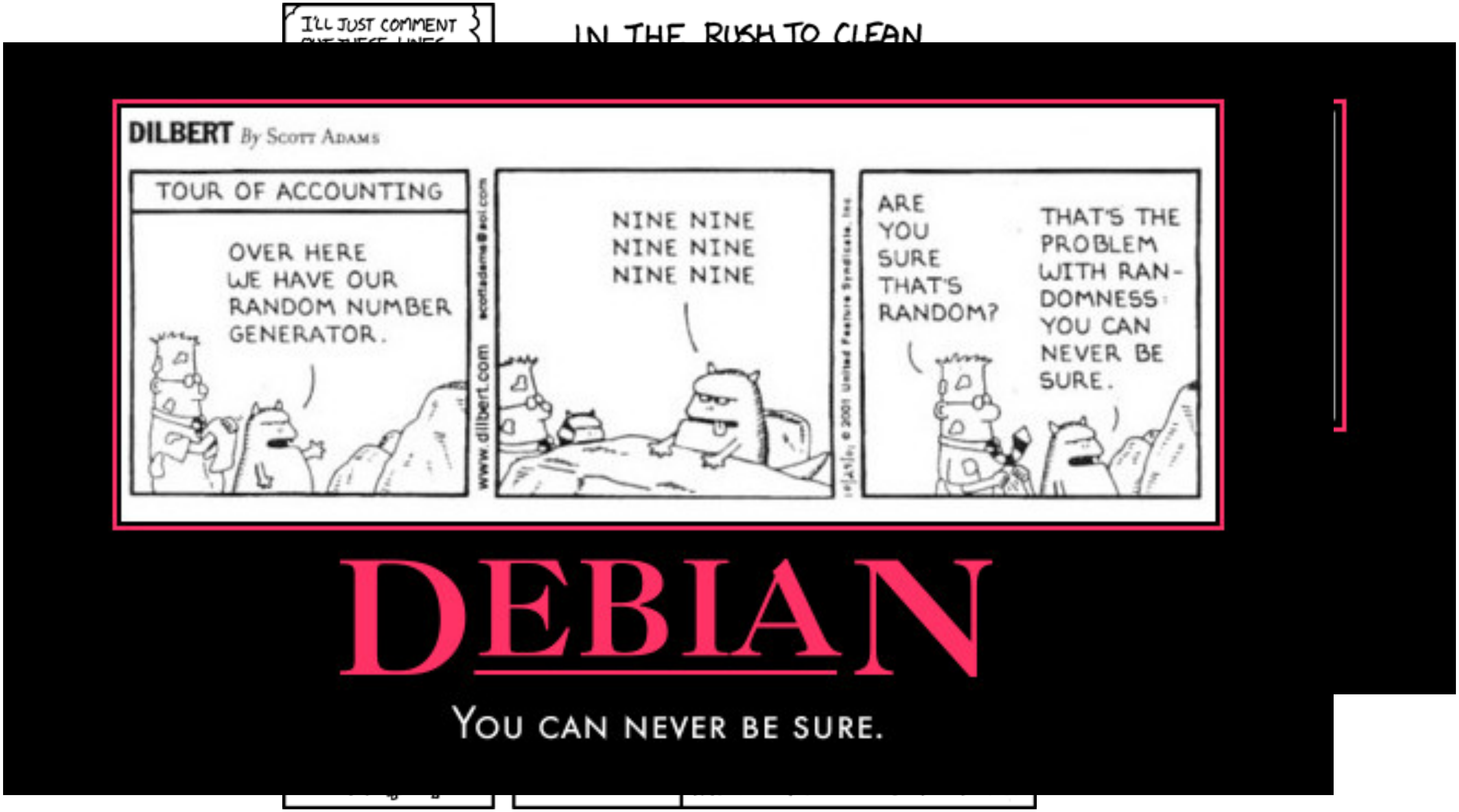
    ```
    MD_Update(&m,buf,j);
    [ .. ]
    MD_Update(&m,buf,j); /* purify complains */
    ```

  - These lines caused Valgrind and Purify to complain (i.e., warnings)
  - Removing the code caused the crippling of the seeding process for OpenSSL

Random Number Bug in Debian

- What does this bug mean?
  - It means that public / secret keys generated after the bug was introduced are not as random as one might think
  - In fact, there are lists on the Internet that one can now download
  - With these, it is possible to decode SSL traffic, login to a remote account, etc.
  - http://www.metasploit.com/users/hdm/tools/debian-openssl/

Random Number Bug in Debian


I'LL JUST COMMENT
IN THE RUSH TO CLEAN
DILBERT By Scott Adams
TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.
www.dilbert.com
scottadams@aol.com
© 2001 United Feature Syndicate, Inc.
DEBIAN
YOU CAN NEVER BE SURE.

General Windows Security

Windows

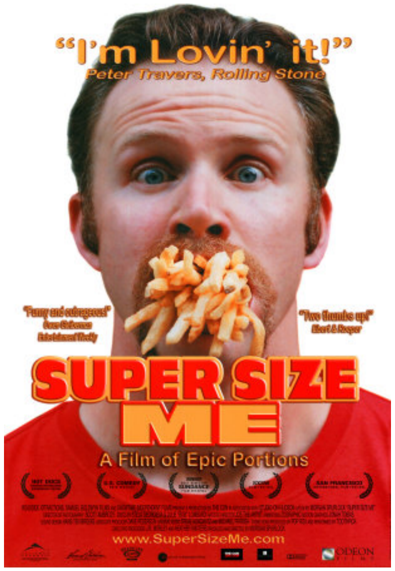
- A lot of computers run Windows. Windows is all around you
  - When dealing with security issues, it is important to have knowledge of Windows.
  - Windows is the best example of non-open source system and security issues.
- Windows security is always in the news (major virus, worm and trojan outbreaks in the past, trojans recently were on Windows). Why?
- Seeing the need (finally), Microsoft started a major initiative for security about 10 years ago
  - Attacks are common, e.g., http://www.windows2000test.com

Reinventing the wheel?

- Development as non-administrator (“Great” idea ;-))
  - Default configuration on Windows system = admin!
  - Principle of *least privilege*
  - Administrator command shell using **runas.exe** (i.e., su -)
  - Store configuration and user information under \HKEY_CURRENT_USER
  - Run services under a restricted user (locking down)
  - Take care in giving debugging privileges
- *I Love You* and *Nimda* would not have worked if computer did not run as **admin**.

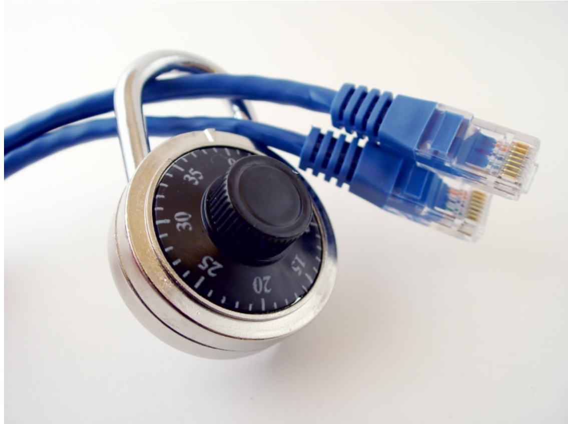
Code size (Windows vs. Linux)

- 1992 Windows 3.1 (3M)
- 1995 Windows 95 (15M)
- 1998 Windows NT 4 (20M)
- 1999 Windows 2000 (40M)
- 2000 Red Hat 6.2 (17M)
- 2000 Debian GNU/Linux 2.2 (55M)
  - Linux 2.2 kernel (1.78M)
  - XFree86 3.3.6 (1.27M)
- 2001 Red Hat 7.1 (30M)



Security at Microsoft

- Trustworthy Computing
  - Windows security push
  - Lead for improved security
- What is it?
  - Training, code reviews
  - Threat models and security testing
- SD3 Security Framework
  - Mind setting
  - Principles to adhere strictly

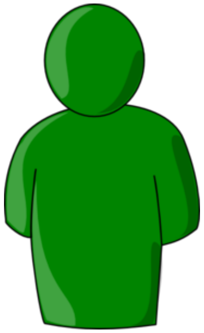


Service Packs and Updates

- Hotfix
  - Single issue / small number of issues
- Security rollup package
  - Single package
  - Multiple hotfixes
- Service pack
  - Major updates
  - Cumulative set of previous updates
  - (optional) Previously *unannounced* fixes
  - (optional) Feature changes
- Major problem: Often **rebooting** is required!

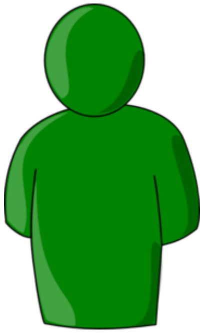
Single User OS (Windows 95/98)

- Almost no security (just like DOS)
  - Anyone can install anything, locking down not possible
- Local Security
  - Highly vulnerable to viruses and trojan horses
  - Highly vulnerable to unauthorized local access/console
  - No file encryption (e.g., like in WinXP).
- Remote Security
  - Highly vulnerable to denial-of-service (weak TCP/IP stack)
    - ping of death, winnuke, land attack
  - If file/print sharing is used
    - Registry can be accessed
  - Win95/98 are not supported by Microsoft anymore (no online updates). There are “zillion” vulnerabilities meanwhile!



Windows 95/98

- Registry
  - used to store system configuration (read/write for all)
- Login Process
  - no authentication – simply press `cancel`
  - determine only profile, don‘t enforce restrictions
- Profile
  - desktop preferences
  - access to saved passwords (in .pwl files)
    - access shared resources, dial-up network
    - Resource Record – Triple <`type, name, passwd`>
    - `passwd` is encrypted with login password



Windows 95/98

- Password files
  - login password is not stored encrypted, instead
  - pwl-file is decrypted with login password and a checksum verified (using user name as well)
  - Windows 95 – algorithm very easy to crack
  - Windows 98 – stronger algorithm (RC4)
    - world-readable
    - vulnerable to brute force / dictionary attacks
  - passwords are always converted to uppercase (makes brute force attacks much easier)
  - unreliable caching mechanism (important information maybe cached)

Windows 95/98 Attacks

- Screen-Saver protection
  - Ctrl-Alt-Del
  - CD-ROM autorun feature to execute programs
    - `autorun.inf` and entry "`open=progname`"
  - Password is stored in Registry
- Malicious Code / Remote exploits
  - 2004 Internet Explorer vulnerabilities (not patched on Win95)
  - Zillion spyware programs, publicly available exploits
  - Good idea not to use Win95/98 – but this is not always possible