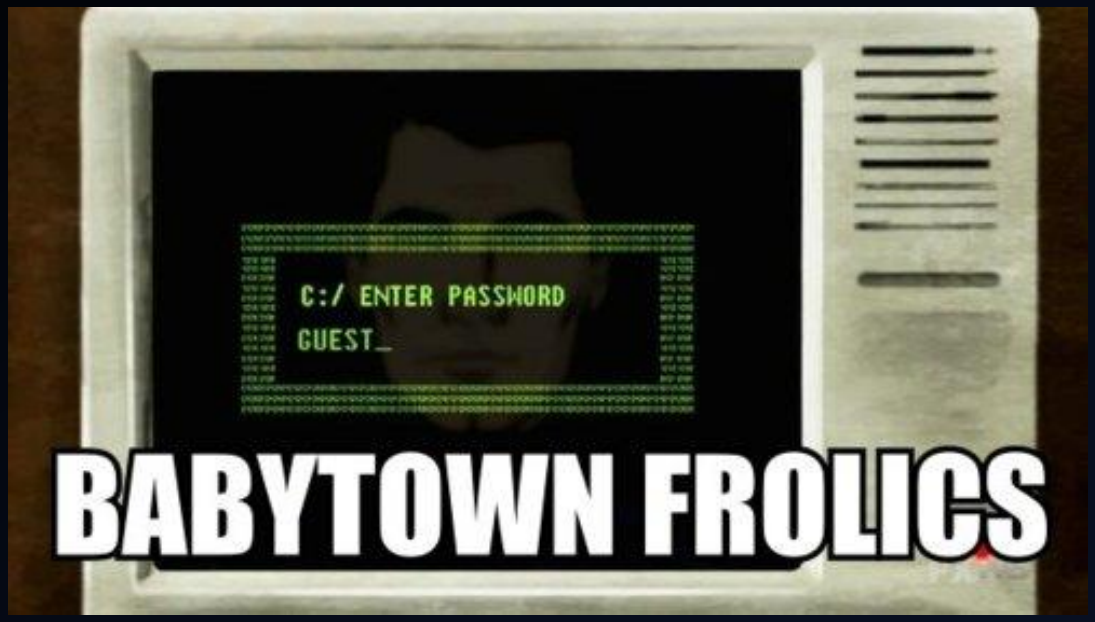
Server Hardening/Defense

LINUX

Presenter: Andrew Folloder

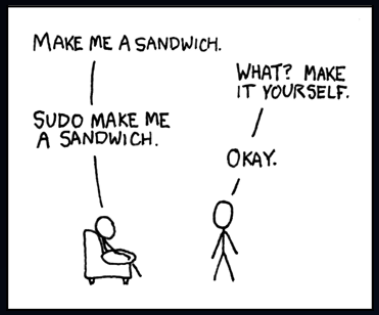
Outline

- First Steps
- Linux Kernel Patches
- Encryption
- Locking Down Networking
- Monitoring



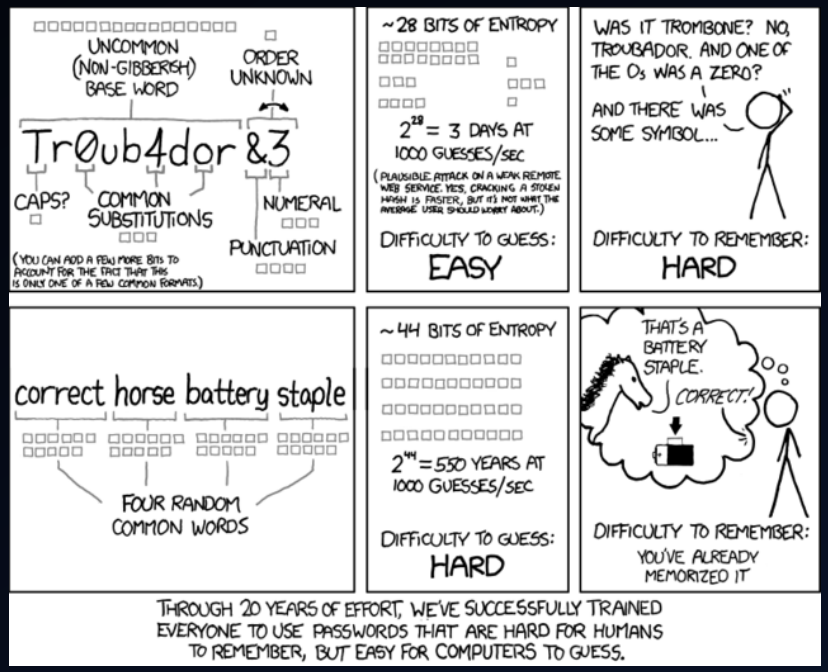
First Steps

- Disable Root Login!!! – use sudo
  - *echo > /etc/securetty*
  - SSH (later…)
- Assign Users Least Privileges
  - Disable Shell access to users that don't need it (ftp, mail users, etc) by changing to /bin/noshell in the /etc/passwd file
  - Have a group for standard users that has limited permissions
    - make sure to block access to tools that can be used to download malicious software like wget, ftp, lynx, etc.
  - Jail users to their home directory via chroot and OpenSSH (alternative makejail/Jailkit)
    - create group for chroot users and add users
      - *groupadd sshusers*
      - *adduser -G sshusers user* OR *usermod -G sshusers user*
    - setup chroot environment (base dir needs to be owned by root)
      - *mkdir -p /jail/{dev,etc,lib,usr,bin}*
      - *mkdir -p /jail/usr/bin*
      - *chown root:root /jail*
      - *mknod -m 666 /jail/dev/null c 1 3*
    - copy over binaries you want the users to have access to
      - *cp -p /usr/bin/bash /jail/bin/*
      - *cp -p /usr/bin/ls /jail/bin/*
    - add the shared libraries needed by the binaries
      - *ldd*
      - l2chroot (http://www.cyberciti.biz/files/lighttpd/l2chroot.txt)
    - configure SSH
      - edit /etc/ssh/sshd_config: *Match group sshusers*
        ChrootDirectory /var/jail/
        X11Forwarding no
        AllowTcpForwarding no




```
/$ ldd /bin/ls
        /usr/lib/arm-linux-gnueabihf/libcofi_rpi.so (0xb6f14000)
        libselinux.so.1 => /lib/arm-linux-gnueabihf/libselinux.so.1 (0xb6ee5000)
        librt.so.1 => /lib/arm-linux-gnueabihf/librt.so.1 (0xb6ed6000)
        libacl.so.1 => /lib/arm-linux-gnueabihf/libacl.so.1 (0xb6ec7000)
        libgcc_s.so.1 => /lib/arm-linux-gnueabihf/libgcc_s.so.1 (0xb6e9f000)
        libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6d70000)
        /lib/ld-linux-armhf.so.3 (0xb6f20000)
        libdl.so.2 => /lib/arm-linux-gnueabihf/libdl.so.2 (0xb6d65000)
        libpthread.so.0 => /lib/arm-linux-gnueabihf/libpthread.so.0 (0xb6d46000)
        libattr.so.1 => /lib/arm-linux-gnueabihf/libattr.so.1 (0xb6d39000)
```

Relevent xkcd



- Minimize Software
  - *dpkg –list*
  - *dpkg --info packageName*
  - *apt-get remove packageName*
- Keep Software Updated
  - *sudo apt-get update && sudo apt-get upgrade*
- User Account & Password Policy
  - Aging: *chage -M 60 userName* (/etc/login.defs)
  - Check user passwords against a dictionary attack
    - *sudo apt-get install libpam-cracklib*
    - add to /etc/pam.d/common-password: *password required pam_cracklib.so retry=2 minlen=8 difok=3*
      - *dcredit=N : Digits characters*
      - *ucredit=N : Upper characters*
      - *lcredit=N : Lower characters*
      - *ocredit=N : Other characters*
  - Limit Password Reuse
    - append *remember=10* to existing password line (e.g. password sufficient pam_unix.so use_authtok md5 shadow remember=10)
  - Lock account after failed login attempts (using pam_tally and faillog)
    - *auth required pam_tally.so no_magic_root*
    - *account required pam_tally.so deny=3 no_magic_root lock_time=86400*
  - Lock accounts with empty passwords
    - *sudo awk -F: '($2 == "") {print}' /etc/shadow*
    - Lock account: *passwd -l accountName*
  - Make sure only root has UID set to 0
    - *sudo awk -F: '($3 == "0") {print}' /etc/passwd*

- Disable Unwanted Services (alternative: sysvconfig)
  - list status of all services: *service --status-all*
  - disable services: *update-rc.d serviceName disable*
- Remove/Disable Unsafe Services
  - FTP, Telnet, Rlogin, Rsh, etc.
- Check all files with root SUID or SGID executables
  - *sudo find / -type f \(-perm /4000 -a -user root \) -ls -o \(-perm /2000 -a -group root \) –ls*
- Separate Disk Partitions
  - create separate partitions for user modifiable directories and block write, execute, and suid/sgid access
    - /usr
    - /home
    - /var and /var/tmp
    - /tmp
  - edit /etc/fstab (e.g. */dev/sda5 /ftpdata ext3 defaults,nosuid,nodev,noexec 1 2*)
- Harden sysctl.conf
  - used to configure kernel parameters at boot time
  - http://www.cyberciti.biz/faq/linux-kernel-etcsysctl-conf-security-hardening/
- Turn off IPv6
  - Edit /etc/modprobe.d/aliases
  - Replace *alias net-pf-10 ipv6* with *alias net-pf-10 off*
    alias ipv6 off

Linux Kernel Patches

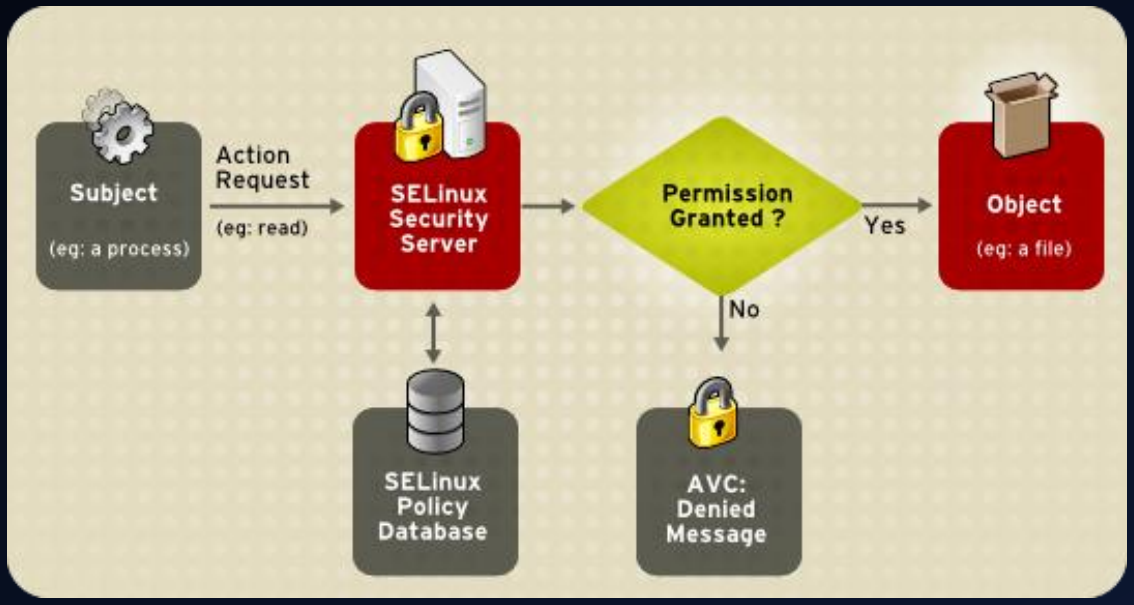
LSM & GRSECURITY

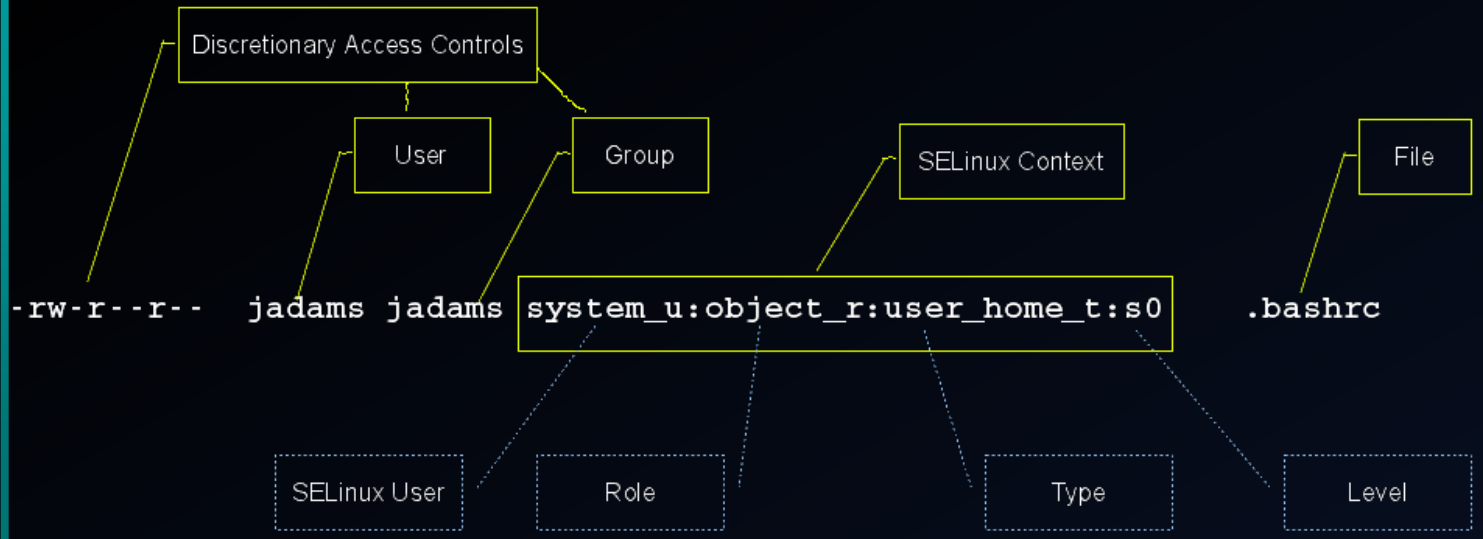
Linux Security Modules (LSM)

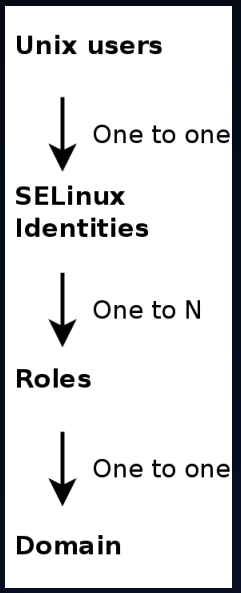
- Framework that allows the Linux kernel to support a variety of computer security
- Designed to provide the specific needs of everything needed to successfully implement mandatory access control (MAC)

SELinux (Security-Enhanced Linux)

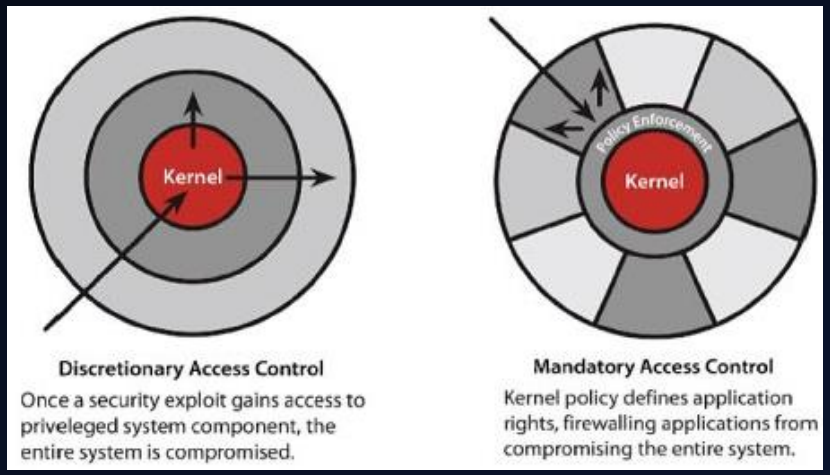
- Developed by NSA, uses LSM to implement MAC (on top of DAC)
- Not a distro, but rather kernel modifications
- Included in CentOS, RHEL, Fedora, Debian, Ubuntu, Suse, Slackware, and more







- Security Context
  - All subjects and objects have a security context (domain -> subjects, file context -> objects)
  - user: SELinux user (not the same as the Linux user) assigned to the resource. Doesn't change (opposed to how sudo works)
  - role: SELinux role in which the resource currently works (e.g. unprivileged user, web administrator, database administrator, etc.)
  - type: Attribute of Type Enforcement that defines a domain for processes, and a type for files.



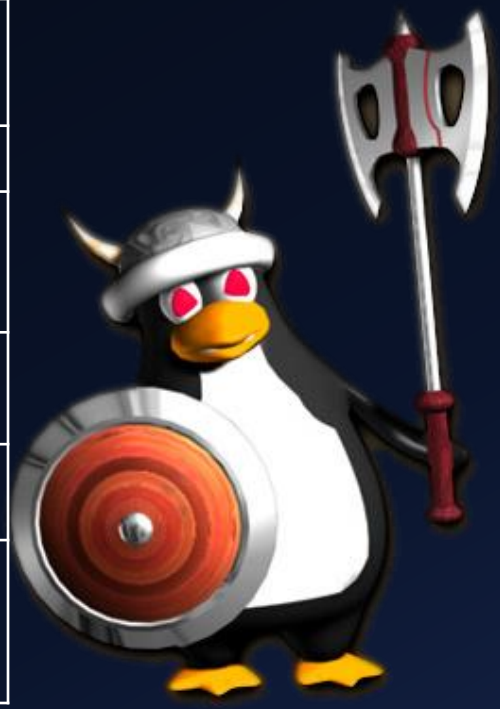
AppArmor

- Created as alternative to SELinux by Novell (under GPL)
- Included in OpenSUSE and Ubuntu
- Very similar to SELinux, but much easier to configure and use

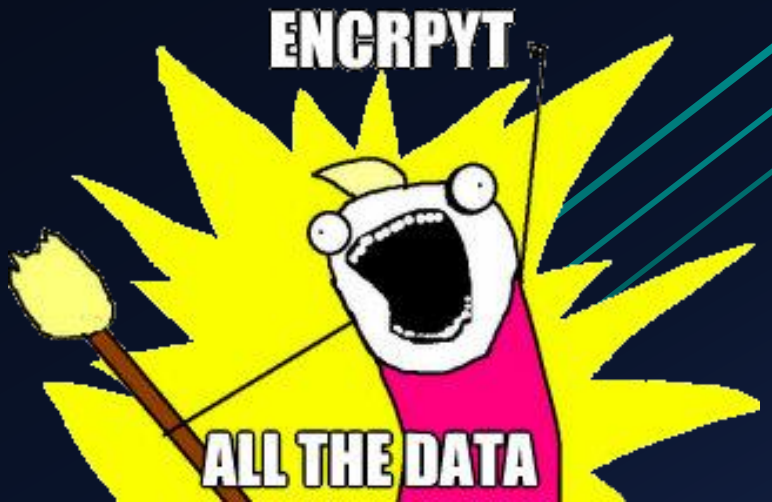
grsecurity

- Provides PaX, Role-based access control (RBAC), Chroot hardening, TPE, and more
- Easiest to use

Feature	SELinux	AppArmor	grsecurity
Automated	No (audit2allow and system-config-selinux)	Yes (Yast wizard)	Yes (auto training)
Powerful policy setup	Yes (very complex)	Yes	Yes
Default and recommended integration	CentOS / RedHat / Debian	Suse / OpenSuse	Any Linux distribution
Training and vendor support	Yes (Redhat)	Yes (Novell)	No (community forum and lists)
Recommend for	Advanced user	New / advanced user	New users
Feature	Pathname based system does not require labelling or relabelling filesystem	Attaches labels to all files, processes and objects	ACLs



Encryption

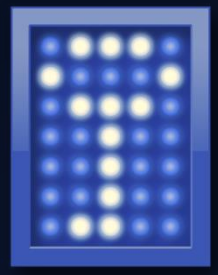


- Use SCP, SSH, or SFTP to transfer files!
- Chroot SFTP users
- OpenSSH
  - config file: /etc/ssh/sshd_config
    - Disable root Login via SSH: *PermitRootLogin no*
    - Change Default SSH Port: *Port 300*
    - Only use SSH Protocol 2: *Protocol 2*
    - Disable .rhosts Files: *IgnoreRhosts yes*
    - Explicetly allow users: *AllowUsers root vivek jerry*
    - Disable Host-Based Authentication: *HostbasedAuthentication no*
  - Use Public Key Based Authentication
  - OpenSSH GateKeeper (Multi factor authentication)
    - https://calomel.org/openssh_gatekeeper.html
- GnuPG
  - Allows you to encrypt and sign your data and communication
  - Generate keys: *gpg --gen-key*
  - Sign file: *gpg --output file.sig --sign file* (compresses, then signs)
  - Encrypt file: *gpg* --encrypt *--recipient 'Your Name' foo.txt*
  - Import a key: *gpg --import key.asc*
  - Search server for key: *gpg --search-keys 'friend@a.com' \ --keyserver hkp://keys.pgp.net*

GnuPG

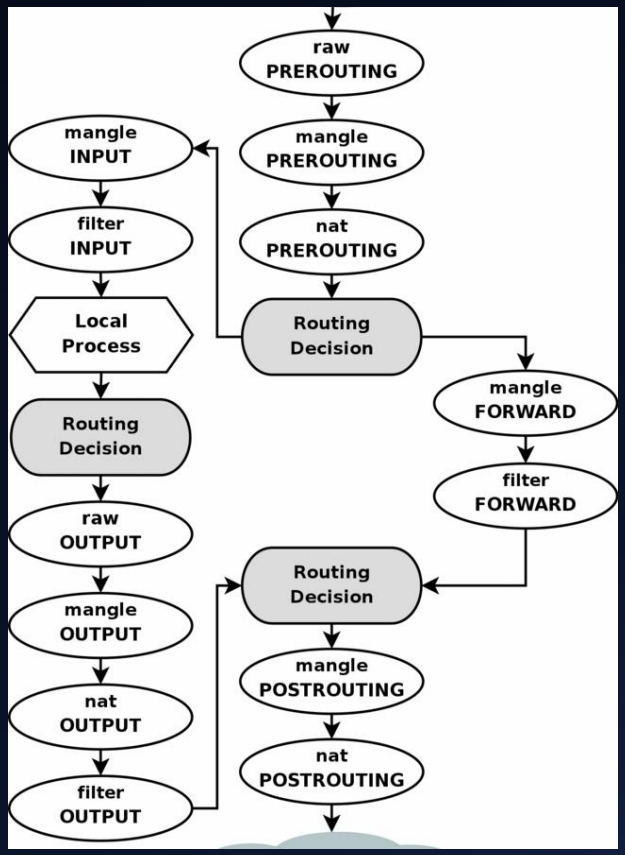
Disk Encryption

- Stacked (System-Level)
  - eCryptfs (default for Ubuntu $HOME)
    - Stores cryptographic metadata in the header of each file written, so that encrypted files can be copied between hosts
    - *sudo apt-get install ecryptfs-utils*
    - *sudo mount -t ecryptfs /ecrypt /ecrypt*
  - EncFS
    - FUSE (Filesystem in Userspace) based
    - Encrypted file metadata kept separately in a central directory (single point of failure)
    - *sudo apt-get install encfs*
    - *encfs /encrypted /decrypted*
    - *fusermount -u /decrypted*
- Block (Device-Level)
  - Truecrypt
    - Need to download through Truecrypt's website (license shenanigans)
    - Great performance and cross-platform support
  - dm-crypt w/ LUKS (Linux Unified Key Setup)
    - Built into Linux kernel, can encrypt whole disks, removable media, partitions, software RAID volumes, logical volumes, and files.
    - *cryptsetup -y -v luksFormat /dev/sdb* (!!!PARTITON WILL BE FORMATTED!!!)
    - *cryptsetup luksOpen /dev/sdb backup*
    - *cryptsetup luksClose backup*



Lock Down Network Services

- Chroot Apache
  - old-fashioned method (manually)
  - mod_security way (simple, but with caveats)
- *IPTables (https://help.ubuntu.com/community/IptablesHowTo)
  - Clear existing rules: *iptables -F*
  - Allow established connections: *sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT*
  - Allow incoming SSH: *iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT*
    iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
  - Set Default Chain Policies: *iptables -P INPUT DROP* (alternative: explicit rule at end of chain)
    iptables -P FORWARD DROP
    iptables -P OUTPUT DROP
- *mod_security
  - https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project
- mod_evasive
  - Prevent DOS, DDOS, Brute Force attacks
- TCPWrapper
  - Host-based networking access control list (ACL) system
- fail2ban
  - Scans log files and bans IPs (via IPTables) based on regex rules
    - can also perform custom actions:
      - email report of event
      - nmap back the attacker (and email the results)



* Covered in our previous topic “Network Security”, so I won’t go into much detail

Monitoring

STAY PARANOID

- Intrusion Detection/Prevention System (IDS/IPS)
  - *Snort
  - OSSEC (HIDS)
    - Host-based IDS that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response.
    - Cross-platform architecture allowing multiple systems to be easily monitored and managed.
    - Meets compliance requirements
    - Agent and agentless monitoring

* Covered in our previous topic “Network Security”, so I won’t go into much detail

OSSEC WebUI
Version 0.3+

Main | Search | Integrity checking | Stats | About

February 08th, 2013 05:41:30 PM

Available agents:

-ossec-server (127.0.0.1)
Name: ossec-server
IP: 127.0.0.1
Last keep alive: 2013 Feb 08 17:41:30
OS: Linux kubuntu 3.5.0-21-generic #32-Ubuntu SMP Tue Dec 11 18:51:59 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux

Latest modified files:

+/usr/bin/kvkbd
+/etc/gimp/2.0/gimprc
+/etc/ssh/sshd_config
+/etc/ssh/ssh_config
+/etc/adduser.conf

Latest events

Level: 5 - **Web server 400 error code.** **2013 Feb 08 17:35:44**
Rule Id: 31101
Location: kubuntu->/var/log/apache2/access.log
Src IP: 127.0.0.1
127.0.0.1 - - [08/Feb/2013:17:35:44 -0800] "GET /announce?peer_id=-KT4300-[redacted]&port=6881&uploaded=0&downloaded=0&left=7864320&compact=1&numwant=200&key=[redacted]&event=started&info_hash=[redacted] HTTP/1.1" 404 489 "-" "KTorrent/4.3.0"

Level: 5 - **Web server 400 error code.** **2013 Feb 08 17:33:22**
Rule Id: 31101
Location: kubuntu->/var/log/apache2/access.log
Src IP: 127.0.0.1
127.0.0.1 - - [08/Feb/2013:17:33:21 -0800] "GET /announce?peer_id=-KT4300-[redacted]&port=6881&uploaded=0&downloaded=0&left=0&compact=1&numwant=200&key=[redacted]&event=started&info_hash=[redacted] HTTP/1.1" 404 499 "-" "KTorrent/4.3.0"

Level: 5 - **Web server 400 error code.** **2013 Feb 08 17:29:26**
Rule Id: 31101
Location: kubuntu->/var/log/apache2/access.log
Src IP: 127.0.0.1
127.0.0.1 - - [08/Feb/2013:17:29:25 -0800] "GET /announce?peer_id=-KT4300-[redacted]&port=6881&uploaded=0&downloaded=0&left=0&compact=1&numwant=200&key=[redacted]&event=started&info_hash=[redacted] HTTP/1.1" 404 489 "-" "KTorrent/4.3.0"

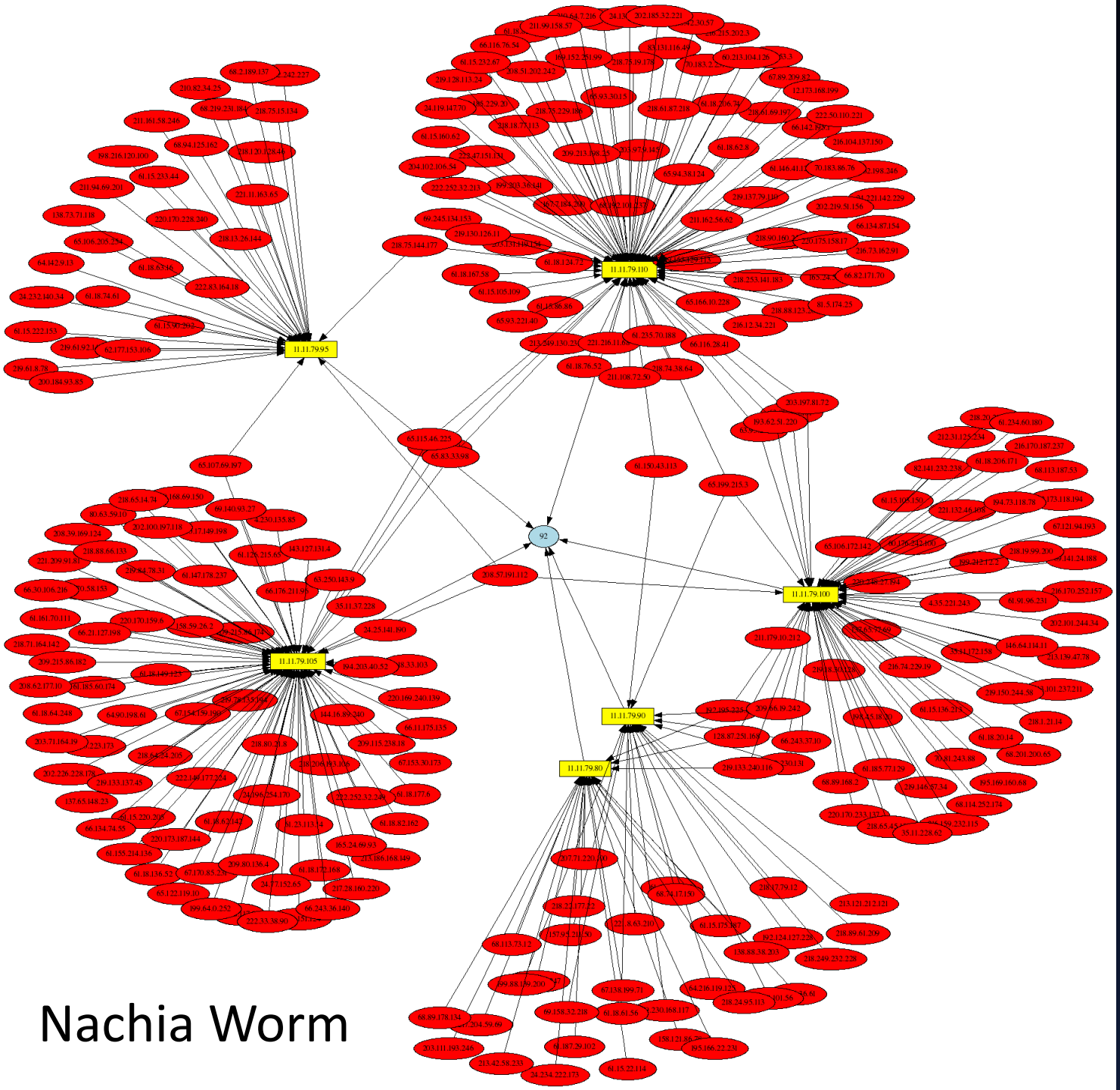
Level: 5 - **Web server 400 error code.** **2013 Feb 08 17:18:47**
Rule Id: 31101
Location: kubuntu->/var/log/apache2/access.log
Src IP: 127.0.0.1
127.0.0.1 - - [08/Feb/2013:17:18:45 -0800] "GET /announce?peer_id=-KT4300-[redacted]C&port=6881&uploaded=0&downloaded=0&left=0&compact=1&numwant=200&key=[redacted]&event=started&info_hash=[redacted] HTTP/1.1" 404 489 "-" "KTorrent/4.3.0"

Level: 5 - **Web server 400 error code.** **2013 Feb 08 17:18:47**
Rule Id: 31101
Location: kubuntu->/var/log/apache2/access.log
Src IP: 127.0.0.1
127.0.0.1 - - [08/Feb/2013:17:18:45 -0800] "GET /announce?peer_id=-KT4300-[redacted]&port=6881&uploaded=0&downloaded=0&left=0&compact=1&numwant=200&key=[redacted]&event=started&info_hash=[redacted] HTTP/1.1" 404 489 "-" "KTorrent/4.3.0"

- Logwatch
  - Basic analysis and display formatting for a wide range of log file types
  - Easy to install and use – works right out of the package on almost all systems

```
# logwatch --service sshd --range=Today --detail=High

 -------------------- SSHD Begin ------------------------

 Illegal users from:
    192.168.1.83: 12 times
       bob/password: 6 times
       george/password: 3 times
       raphael/password: 3 times

 **Unmatched Entries**
 pam_succeed_if(sshd:auth): error retrieving information about user raphael :
3time(s)
 pam_succeed_if(sshd:auth): error retrieving information about user bob : 6
time(s)
 PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser=
rhost=192.168.1.83   : 4 time(s)
 pam_succeed_if(sshd:auth): error retrieving information about user george : 3
time(s)

 --------------------- SSHD End -------------------------
```

- psad
  - Collection of daemons that analyze iptables log messages to detect port scans and other suspicious traffic
  - Incorporates signatures from Snort to detect probes for backdoor programs, DDoS tools, advanced port scans
  - Passively fingerprint remote operating systems from which scans originate
  - Forensics mode iptables logfile analysis
  - Configurable scan thresholds and danger level assignments
  - Parsing of iptables log messages and generation of CSV output that can be used as input to AfterGlow


11.11.79.110
11.11.79.95
11.11.79.105
11.11.79.100
11.11.79.90
11.11.79.80
92

Nachia Worm

- auditd
  - userspace component to the Linux Auditing System
  - rules in /etc/audit.rules are read at startup
  - audit the /etc/passwd file: *auditctl -w /etc/passwd -p war -k password-file*
  - file system audit rule: *auditctl -w /tmp -p e -k webserver-watch-tmp*
  - syscall audit rule using pid: *auditctl -a entry,always -S all -F pid=1005*
  - *ausearch -f /etc/passwd*
  - *aureport -ts today*

BONUS

- Bastille (http://bastille-linux.sourceforge.net/)
  - Hardening program that "locks down" an operating system
  - Interactive interface that'll walk you through and explain things as it asks you questions
  - can also assess a system's current state of hardening, granular reporting on each of the security settings with which it works.
  - Assessment Report Criteria