Basic Linux Desktop Security

Think Security: 5Q

1)What is the problem?

2)What is the proposed solution?

3)What is the cost?

4)Is it effective?

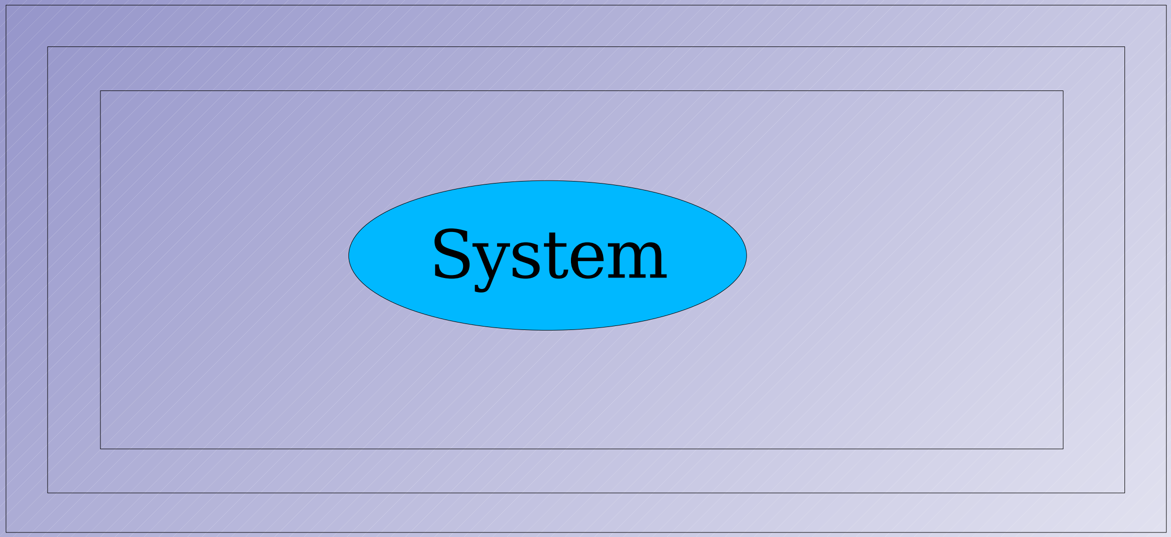
5)What are the collateral damages?

➔ Worth it?

(from: Schneier “Beyond Fear”)

Think Security: SPoF

- Never design a single point of failure!
- Design as layers of security.



Think Security: Trust?

- Design a threat model:
  - What are you concerned about?
  - What components/people do you trust?
  - How far do you trust them?
  - With what do you trust them?
  - Are you sure?

Distribution

- Chose a “distribution” that is designed for what you want to do.
- …one you can trust.
- Find its security mail list.
  - eg. Debian: debian-security-announce (AT) lists.debian.org
- Find out how things are done with it.
- Keep updating! (at least once/week)

Passwords

- Chose secure passwords:
  - mix upper-/lower-case letters, special characters and digits
  - don't use words from any dictionary
  - make it long enough (about 8 characters)
  - eg. use a sentence that you can remember, use the first letter of each word, replace some letters by digits
- Remember: Passwords are unsafe.

Get rid of Junk

- Shutdown/uninstall anything you don't need.
  - Do you need CUPS? (printer?)
  - Do you need Apache? (is this a web-server?)
  - Do you really need FTP?
  - Do you really need five different databases?
  - Do you need GCC? (Are you a developer?)

Configure Securely

- Default config is often insecure.
- Don't allow open mail relay.
- Don't allow open web proxies.
- SSH:
  - Don't open it to outside network if not necessary.
  - Don't use password-authentication.

Restrict Users

- NEVER WORK AS ROOT!!!
- Don't give users rights they don't need.
- Don't give anyone your password.
  There is no excuse!
  No Admin will ever need it!
- Use different passwords on different systems. (Or groups of systems.)

Firewalls

- Every system has a right to its own firewall!
- Never connect an unprotected system to any network.
- Make the firewall as tight as possible.
  - Opening up later is easier than cleaning up an incident.
- Whenever possible: block both directions.
- Don't forget IPv6!

Random Thoughts

- Alteration Detection: Tripwire
- Intrusion Detection: Snort
- Rootkit Scanner: Tiger
- etc.pp.

IP-Tables

- Tables:
  - filter - the actual firewall part
  - nat - network address translation
- filter chains
  - INPUT, FORWARD, OUTPUT
- nat chains
  - PREROUTING, POSTROUTING, OUTPUT

IP-Tables: filtering

- Use tools: iptables-save/-restore
- Policy: DROP/REJECT
- Divide Traffic per Interface.

```
*filter
:INPUT DROP
:FORWARD DROP
:OUTPUT DROP
:netw -
:netout -
:rootout -

#Incoming Traffic
#accept loopback:
-A INPUT -i lo -j ACCEPT
#filter all others
-A INPUT -j netw
#paranoid droppings:
-A INPUT -j DROP
```

```

#This is not a router:
-A FORWARD -j DROP

#Outgoing Traffic
#accept loopback
-A OUTPUT -o lo -j ACCEPT
#filter all others
-A OUTPUT -j netout
#reject remainder
-A OUTPUT -j REJECT
```

IP-Tables: INPUT

- Accept what you requested.
- Accept legitimate users.
- Reject everything else.

```
#accept ICMP
-A netw -p icmp -j ACCEPT

# accept DNS (see query_source in /etc/bin/named.conf)
-A netw -p udp -m udp --sport 53 -j ACCEPT
-A netw -p tcp -m tcp --sport 53 -j ACCEPT
# XNTP
-A netw -p udp -m udp --dport 123 -j ACCEPT
# accept SSH
-A netw -p tcp -m tcp --dport 22 -j ACCEPT

# accept TCP when established from local
## filter SYN
-A netw -p tcp -m tcp --syn -j DROP
## accept TCP above 1024 (denies communication with priv. ports)
-A netw -p tcp -m tcp --dport 1024: -j ACCEPT
```

IP-Tables: OUTPUT

- Allow desired services.
- Allow some users.
- Reject everything else.

```
#Allow ICMP
-A netout -p icmp -j ACCEPT

#Allow DNS in both directions
-A netout -p tcp -m tcp --dport 53 -j ACCEPT
-A netout -p udp -m udp --dport 53 -j ACCEPT
#Allow XNTP
-A netout -p udp -m udp --sport 123 -j ACCEPT
##Owner exceptions
#root
-A netout -p tcp -m owner --uid-owner 0 -j rootout

##Filter TCP SYN
-A netout -p tcp -m tcp --syn -j REJECT
#allow remainder (established) of TCP
-A netout -p tcp -m tcp -j ACCEPT

##Filter all UDP
-A netout -p udp -m udp -j REJECT
```

IP-Tables: users

- Allow only needed services.
- Prevents broken systems from doing more harm.

```
### Root (apt-get)
#update.pureserver.info:
-A rootout -p tcp -m tcp -d 195.20.242.2 --dport 80 -j ACCEPT
#security.debian.org
-A rootout -p tcp -m tcp -d 128.101.240.212 --dport 80 -j ACCEPT
-A rootout -p tcp -m tcp -d 212.211.132.32 --dport 80 -j ACCEPT
-A rootout -p tcp -m tcp -d 212.211.132.250 --dport 80 -j ACCEPT
```

IP-Tables: chain-design

- Policy: don't allow anything.
- Start at bottom: reject everything.
- Move upwards:
  - allow wanted services
  - make exceptions
  - become more specific

Questions?