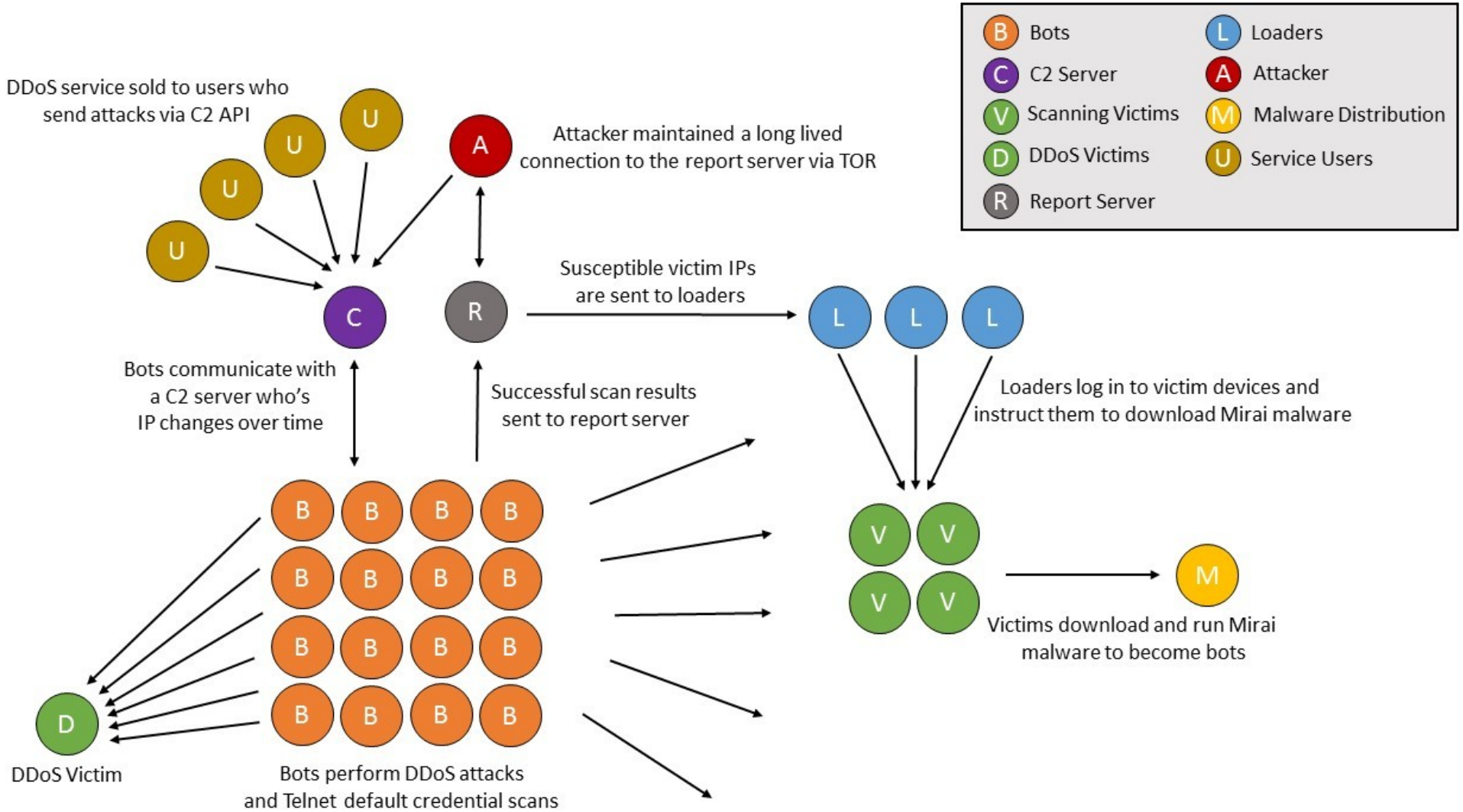
Embedded Many-bit Linux

Quick Primer on Linux Security

Zak Estrada

Mirai – 10s of Millions of Embedded Systems



Source: http://news.softpedia.com/news/there-are-almost-half-of-million-iot-devices-infected-with-the-mirai-iot-malware-509432.shtml

Embedded device security is a big deal

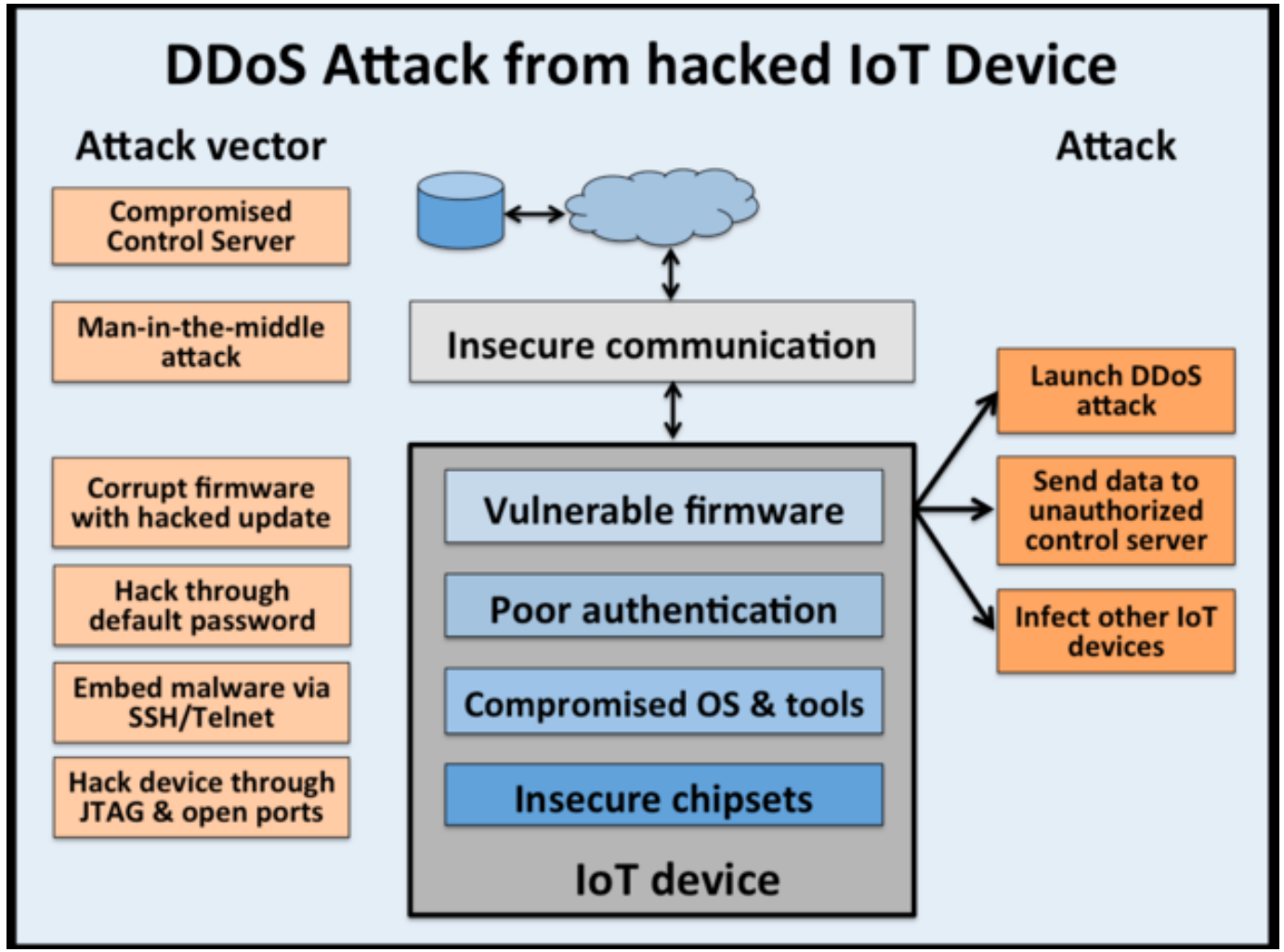
- Embedded systems are often found in critical systems

- Cyber-physical systems
  - Power infrastructure
  - Medical devices
  - Cars

- Embedded devices typically use low-level interfaces
  - Lots of security issues there

Is this a critical system?

Attack Vectors



Source: https://www.networkworld.com/article/3128372/internet-of-things/ddos-attacks-using-iot-devices-follow-the-manchurian-candidate-model.html

Your BeagleBone was designed for ease of use

- "Security comes later"
  - Example: no root password by default!
    - Is this still true?
- If you put it on the internet, it becomes an easy target
- Set a root password
- Only use root/sudo when needed
  - Create users/groups otherwise

Brute-force attack

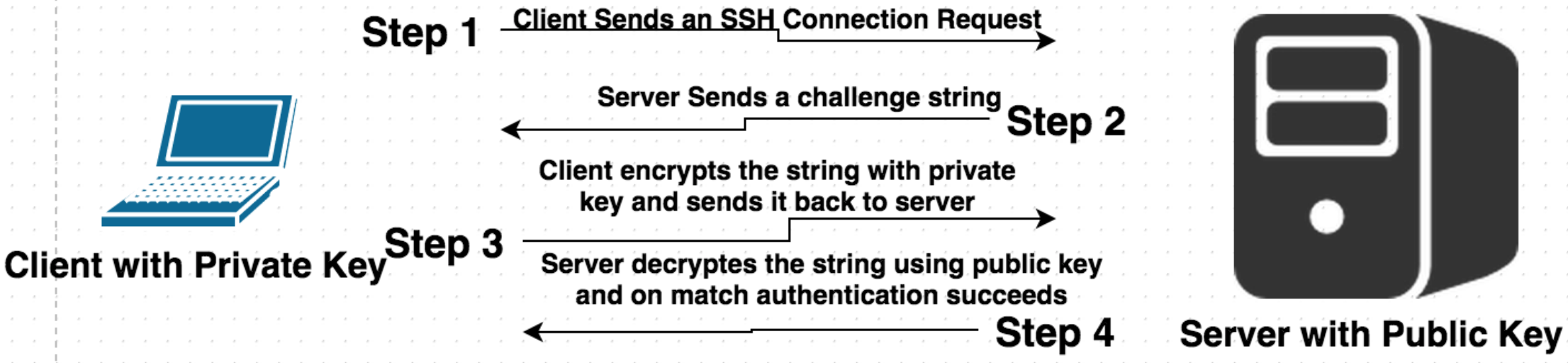
- An attacker can break in by guessing different possible password combinations
- Can also be smarter with a dictionary attack

Preventing brute-force attacks

- Fail2ban
  - Automatically sets up firewall after multiple incorrect passwords
- Disable root ssh
  - /etc/ssh/sshd_config
  - Remember to restart ssh!
- Use a strong password
- Or... use public key authentication

Alternative to password authentication

- SSH public key auth: built on public key cryptography
  - Keys are easy to generate and verify, but difficult to guess



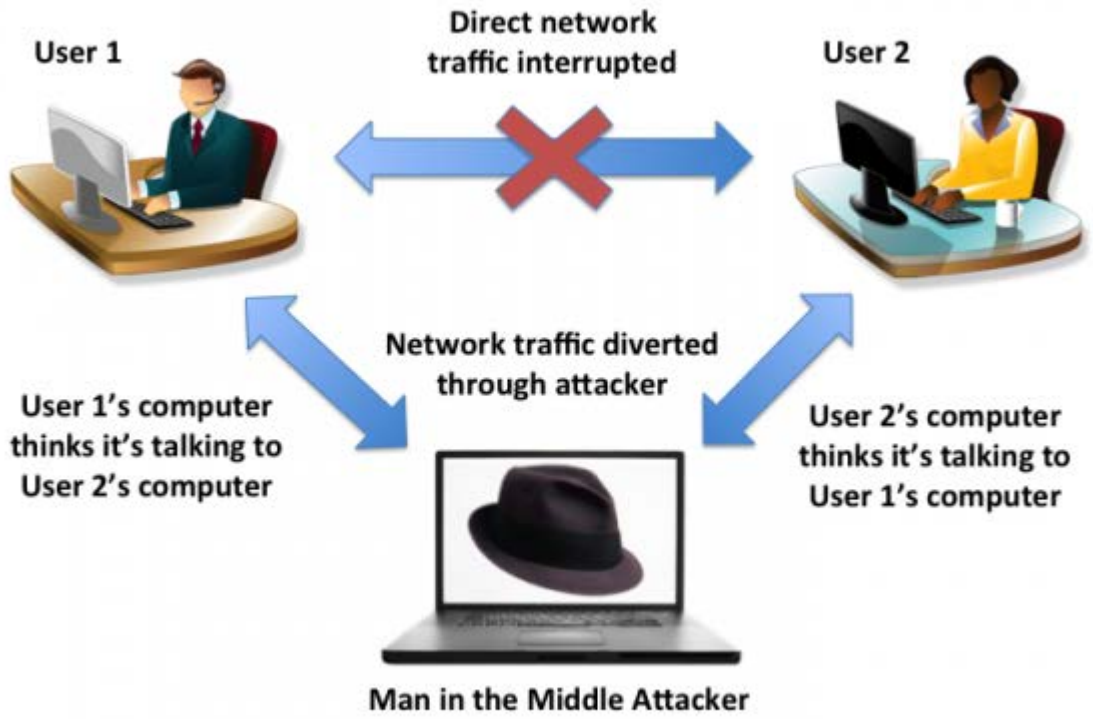
Source: https://vmcentral.zendesk.com/hc/en-us/articles/205576449-How-to-Configure-SSH-to-Accept-Only-Key-Based-Authentication

Using SSH Public Key Authentication

- Generating a key:
  - **ssh-keygen**
  - Can add a passphrase (recommended, look into ssh-agent if you do this)
- Using a key:
  - **ssh-copyid SERVER**
    - Use your password once
    - Look at ~/.ssh/authorized_keys
- It's not uncommon to see servers that don't ever use passwords
  - Cloud instances, etc…

SSH Keys are used for more than auth

- I can use a key to show the host who I am
- But how do I know the host is who **THEY** claim to be?
- *Man in the middle attack*



SSH Host verification

- What's the first thing you do when sshing to a host?

- Can check with **ssh-keygen –l –f *KEYFILE***
  - On the beaglebone
  - E.g, KEYFILE=/etc/ssh/ssh_host_ecdsa_key.pub
  - If this is not your first time connecting you have already saved the host key signature
  - You can check from your VM: **ssh-keygen –l –F *HOSTNAME***

- Those keys should be the same

Principle of Least Privilege

- Every entity must only access the resources needed for its legitimate function and nothing more.
  - AKA, the most restrictive set of permissions

- An entity can be:
  - User
  - Program
  - Process
  - Local or Remote

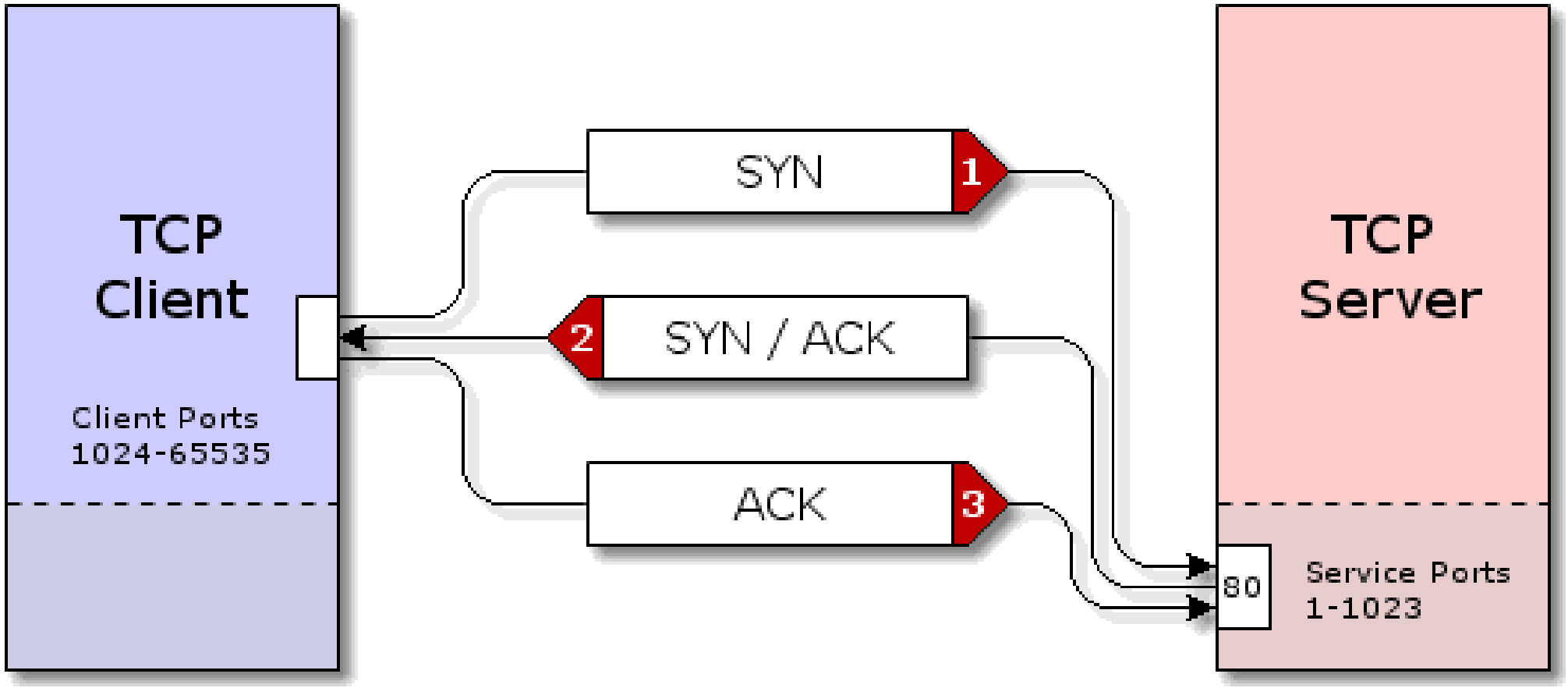
Security auditing

- Auditing is testing your system to ensure that its behavior meets your security needs/expectations

- With local users and files, it can be easy to determine what privileges each user has, either by using commands or by viewing permissions (e.g., **groups USER**, **ls –l**, etc…)

- For network security, it can be quite difficult since there are so many applications use the network

TCP uses a Three-way Handshake



Source: https://tuxawy.files.wordpress.com/2012/04/tcp-connect1.gif

nmap

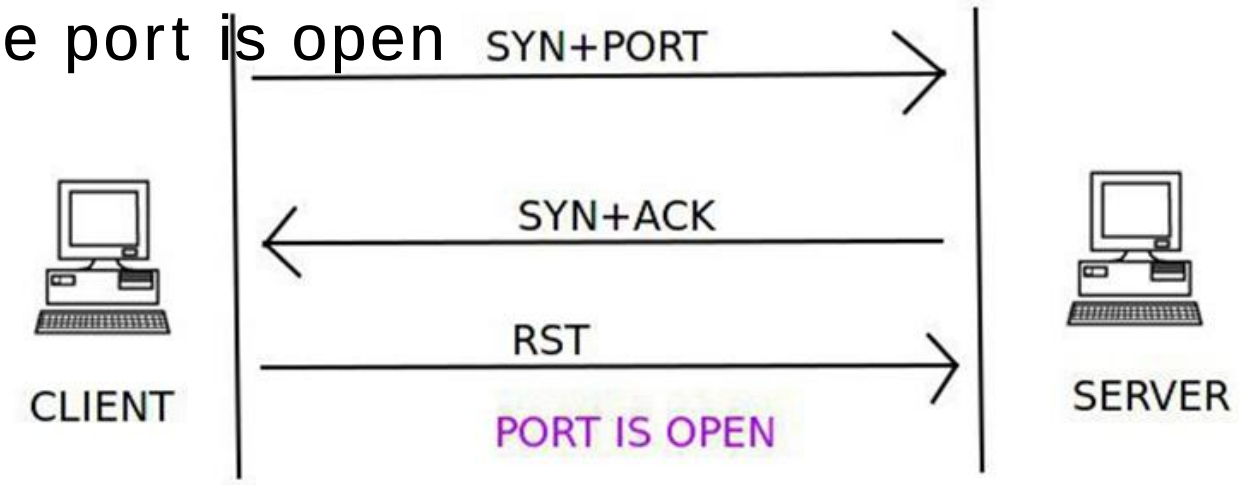
- "Network Mapper"
- Open-source tool for network discovery and security "auditing"
- Mainly used for port scanning
  - i.e., seeing what's listening on the host

We can use nmap to see what ports are open

- **$ nmap –sS HOST**
  - TCP SYN scan
  - The most popular nmap function

- Half-opens connections to do the scan
  - Send a SYN to every port on HOST
  - If you get a SYN/ACK, the port is open

- Requires root



Run nmap –sS against your bone

- From your linux VM
- `sudo apt-get install nmap`
- `nmap –sS bone`

- What do you get as a result?

Running nmap –sS for real

```
zak@HOST:~$ sudo nmap -sS bone

Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-01 15:07 EDT
Nmap scan report for bone (192.168.7.2)
Host is up (0.0079s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3000/tcp open  ppp
9090/tcp open  zeus-admin
MAC Address: C8:A0:30:B7:F9:71 (Texas Instruments)

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
zak@HOST:~$
```

What if I forgot the IP address of my bone?

- I could use nmap to find it
- Let's say I remember the beaglebone is on the 192.168.7.0/24 network
- I can scan it with **nmap –n –PS22 192.168.7.0/24**
  - -n: no DNS resolution
  - -PS22: Parallel TCP SYN ping on Port 22
  - Could add **–A** for OS detection, version detection, script scanning and traceroute

nmap –n –A –PS22 192.168.7.0/24

```
Nmap scan report for 192.168.7.2
Host is up (0.0092s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp   open  http?
|_http-methods: No Allow or Public header in OPTIONS response (status code 404)
|_http-title: Introduction to BeagleBoard.org
3000/tcp open  ppp?
9090/tcp open  zeus-admin?
3 services unrecognized despite returning data. If you know the service/version,
 please submit the following fingerprints at http://www.insecure.org/cgi-bin/ser
vicefp-submit.cgi :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=6.40%I=7%D=11/1%Time=5818F453%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,F01,"HTTP/1\.1\x20200\x20OK\r\nX-Powered-By:\x20Express\r\nAccep
SF:t-Ranges:\x20bytes\r\nCache-Control:\x20public,\x20max-age=0\r\nLast-Mo
SF:dified:\x20Fri,\x2008\x20Jul\x202016\x2020:53:08\x20GMT\r\nETag:\x20W/\
SF:"de0-155cc490f20\"\r\nContent-Type:\x20text/html;\x20charset=UTF-8\r\nC
SF:ontent-Length:\x203552\r\nDate:\x20Tue,\x2001\x20Nov\x202016\x2020:00:2
SF:2\x20GMT\r\nConnection:\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html><head
SF:>\n\x20\x20\x20\x20<title>Introduction\x20to\x20BeagleBoard\.org</title
.
```

Okay, great I know what ports are open

- How do I close them off?!?!

- Linux comes with a built in firewall
  - Managed using the “iptables” command
  - iptables using a “chain” of rules that are processed when a network packet arrives
  - Each command adds a rule to the chain

- Excellent description:
  - https://www.globo.tech/learning-center/linux-native-firewall-introduction-to-iptables/

If want to close port 3000 from the outside

- Run two commands:
  - iptables -A INPUT -s 192.168.7.0/24 -p tcp -m tcp --dport 3000 -j ACCEPT
  - iptables –A INPUT –j DROP –p tcp –m tcp –dport 3000 (maybe don't need: -s 192.168.7.0/24)
  - Order is important (-A means append)
- If you want to see which iptables rules are active
  - iptables –L
- To flush the rules and restore default operation
  - iptables -F

iptables are cleared at boot

- `apt-get install iptables-persistent`
- `iptables-save > /etc/iptables/rules.v4`

- HERE BE DRAGONS

What if I want to use those ports sometimes?

- We can either make our rules less restrictive to open up access to more networks
  - Add another “–j ACCEPT” rule before the “-j REJECT” rule

- Alternatively, if we have SSH open, we could do ssh port forwarding
  - You can use SSH port forwarding to connect a local port on your computer to a port on the beaglebone, through SSH
  - BONUS: everything over the tunnel gets encrypted

SSH Port Forwarding

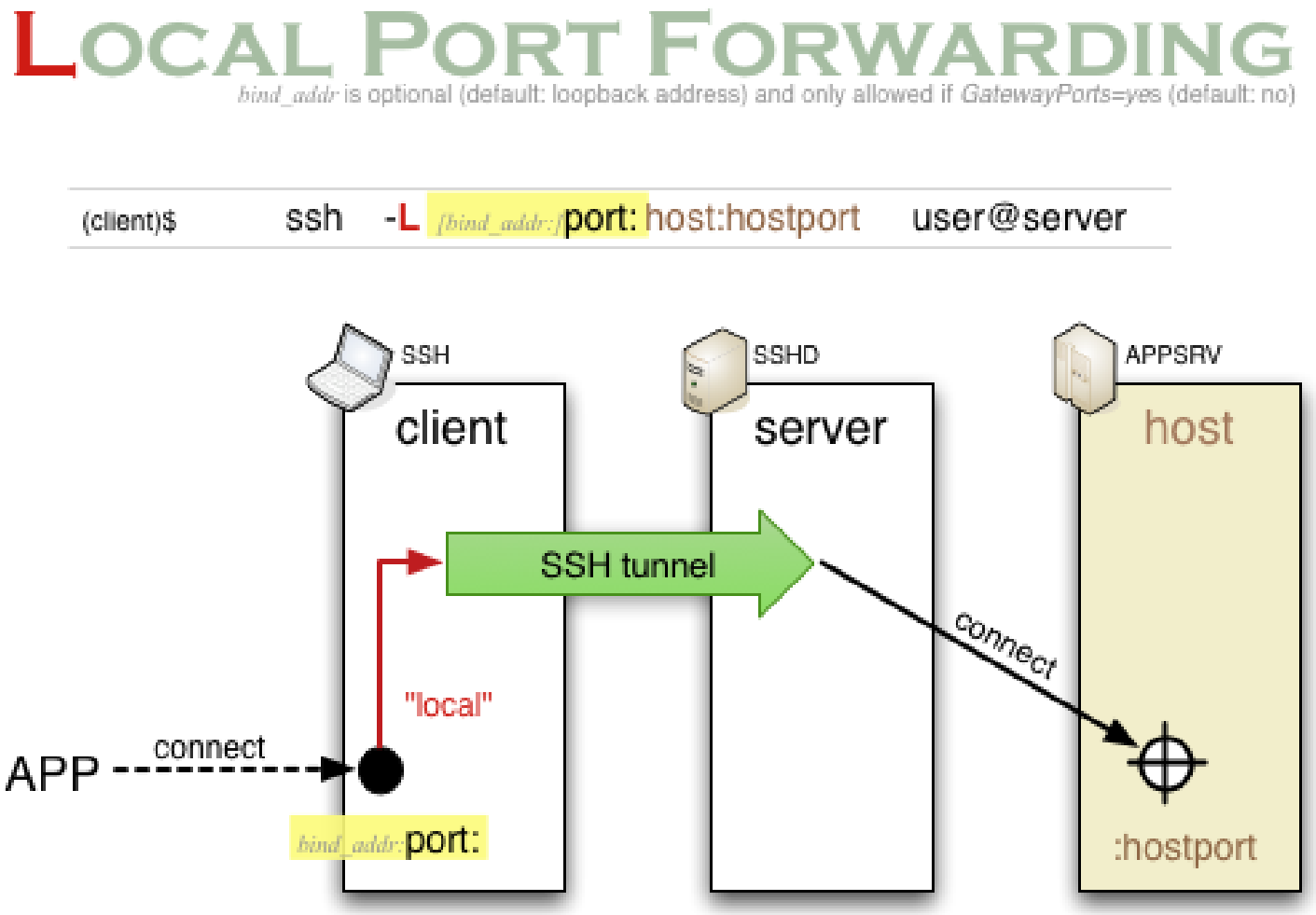


Image source: http://www.dirk-loss.de/ssh-port-forwarding.png

SSH Port Forwarding example

- So, to get to our port 3000, we would execute
  - **ssh bone –L3000:localhost:3000**

- This forwards port 3000 on the beaglebone …
  - We use “localhost” since the command takes the ssh server’s perspective
- and forwards it to port 3000 on the host
  - where you are sshing from

- This means we connect to port 3000 on the host to reach port 3000 on the bone, as long as we keep ssh open

Summary

- Embedded device security
- Principle of least privilege
- TCP handshake
- Port scanning
  - Nmap
- Firewall
  - Iptables
- Port forwarding
  - Using ssh
- Public Key Authentication

Random stuff: let's say I forgot my password

- The linux kernel has arguments just like any other program
- Grub bootloader lets you change the kernel's command line arguments