Services

UBNetDef, Fall 2021

Week 6

Lead Presenter: Lucas Crassidis

WITH HELP FROM, PHIL FOX, STEPHEN JAMES, AIBEK ZHYLKAIDAROV

AN OVERVIEW:

- It's really awesome that you're here!
  - Today we are introducing some concepts about services
  - We're also walking through an assignment very similar to this week's homework together.
- Take very good notes tonight
  - You'll use many similar commands on your homework
  - These slides cover steps. Critical file locations and commands are delivered in-class

YOU GOT SERVED

- In GENERAL:
  - A client: Runs a bunch of services for a limited amount of users
  - A server: Runs a limited amount of services for a whole lotta users

SERVE THE SERVANTS

- The term "Services" can be ambiguous
  - Technically, your laptop's wireless network manager is a "service."
  - What we're after are server-served services. Which ones can you name?

PROTOCOLS

- An agreed-on way to communicate
- What kind of protocols can you name? (Like, real life stuff...)

PROTOCOLS cont.

- An agreed-on way to communicate
- What kind of protocols can you name? (Like, real life stuff…)
- For Machines: Provide a way to store, manage, and access data
- Machines agree on Data types and Ports to transfer data over

PROTOCOL EXAMPLE: DATABASES

- No "standard" ports, DBMSs have their own communication protocols
  - Usually have their own clients to interact with them
- Popular examples:
  - MariaDB/MySQL: 3306/tcp
  - Microsoft SQL Server (MSSQL): 1433/tcp
  - MongoDB: 27017/tcp
  - PostgreSQL: 5432/tcp
  - Redis: 6379/tcp

OTHER SERVICE PROTOCOLS:

- Email: SMTP, POP3, IMAP
- DNS!
- Remote access: RDP (Windows!), SSH
- File transfer: FTP, SCP (SSH)
- Web: HTTP, HTTPS (starting to sound familiar?)
- ...and many more!

Remote VS Local

- Endpoints/Hosts:
  - Clients
    - Windows, Ubuntu Desktop, etc.
    - Host is the PC you are currently on
  - Servers
    - Active Directory
    - DNS
    - Web server
- Local
  - Within the same host and ONLY that host
- Remote
  - Separated by at least one (non-local) network connection

IMPLEMENTATION 1:

- Database Setup on RockyDB:
  - Use netstat to check if SQL is running, It's on port 3306
  - Check the Status of MariaDB, you may need to install it
  - Start the MariaDB Service
  - Enable the Service for Automatic Start
  - Verify that MariaDB is enabled and running
  - Improve the security of MariaDB using mysql_secure_installation
  - Verify that MariaDB is listening on the correct port
  - Verify that the Public Zone is currently active on your RockyDB firewall
  - Permanently whitelist the port in the "public" zone in your RockyDB Firewall

What is a Wiki

- Database sends info to wiki site
- Needs both a web server and database server

DB

Serves:
Database Info

WS

Serves:
Webpages

Webpage

IMPLEMENTATION 2:

- Web Server Setup on Web:
  - Use wget to download MediaWiki
  - cd /tmp/
  - Wget https://releases.wikimedia.org/mediawiki/1.35/mediawiki-1.35.1.tar.gz
  - Extract the archive
  - sudo mkdir /var/lib/mediawiki
  - Move the contents of the extracted mediawiki to var/lib/mediawiki

SERVICES TO FOCUS ON: SSH

- This is a remote access protocol that moves a user from one host to another
- Computer Science assignments may require this protocol for turning assignments in!
- Offers secure communication
  - Typically used to access a shell (via the command line) or to remotely execute a command
  - Among other things, it can also be used to copy files (e.g., SCP, SFTP)
- Standard port: 22/tcp
- OpenSSH is, by far, the most common (and free) SSH client and server service

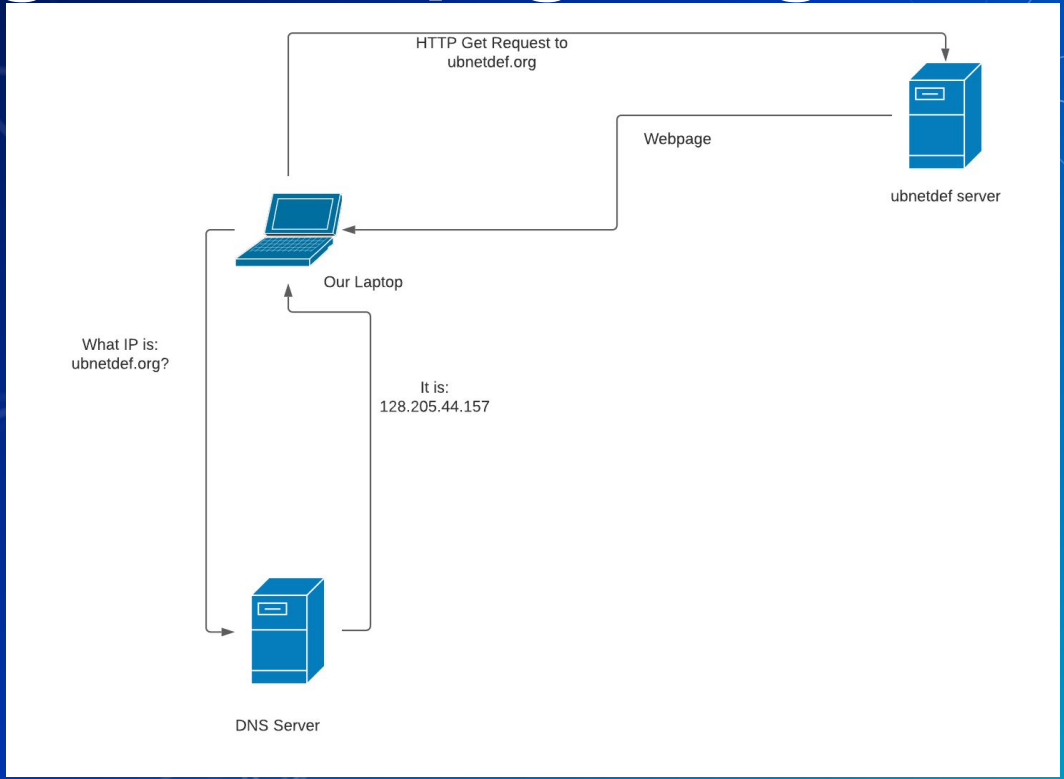
SERVICES TO FOCUS ON: WEB

- Web servers process incoming requests from clients for web resources over HTTP and related protocols
  - Web resources are identified by a Uniform Resource Locator (URL)
  - Might perform additional processing while handling the request
- HTTP is unencrypted; data is transmitted in plaintext
  - Anyone on any of the networks on a path from you to the server can see this data
  - VPNs can obscure this otherwise eavesdroppable data
- HTTPS is an extension of HTTP that is encrypted using TLS, or previously, SSL
  - Client is also able to authenticate the server (using the server's certificate often handled by an authority)

SERVICES IN CONCERT: HOW WE GET TO HTTPS://UBNETDEF.ORG/

- Get an IP address, gateway, etc.
  - Either via DHCP or static IP configuration (network service)
- Resolve "ubnetdef.org" to an IP address
  - Ask a DNS server for the A (IPv4) records for "ubnetdef.org"
  - DNS server should respond with "128.205.44.157"
- Send an HTTP GET request to 128.205.44.157 asking for host ubnetdef.org and path "/"
  - TCP handshake starts, and public keys etc. are exchanged (since we're using HTTPS)
  - Client (browsers etc.) will do
  - Web server processes request then responds
- Note that the above steps are simplified: a lot more happens!

Getting to a Webpage Diagram

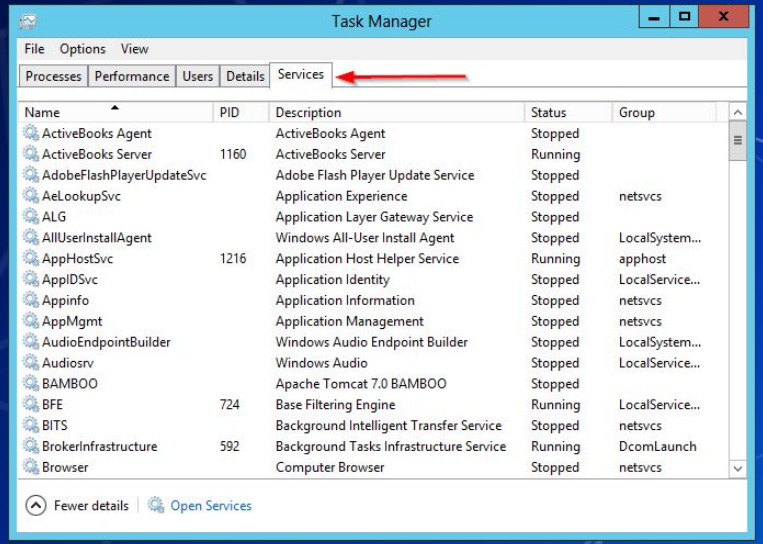


Process VS Service

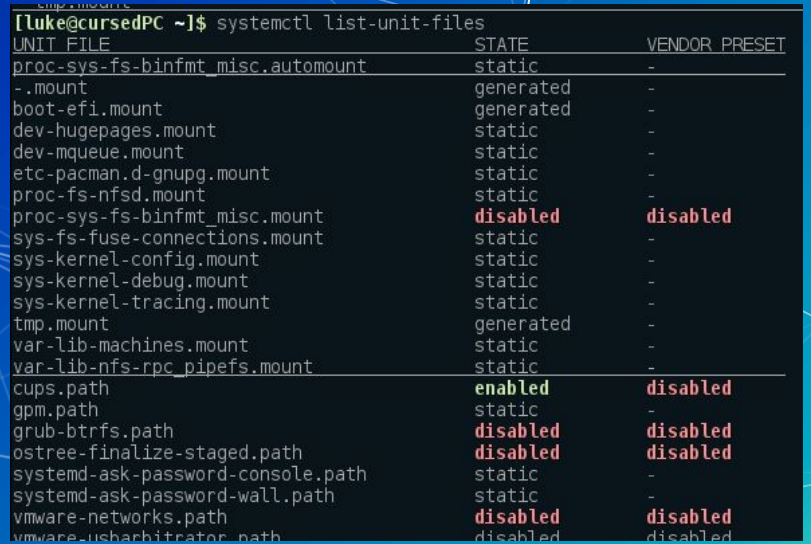
- Process
  - you control when it starts and stop
  - When you click on Firefox, that starts a process
  - When you boot up Minecraft, that is also a process
- Service
  - Continuous and always running when started
  - Needs a command(aka: daemon) to run
  - In Linux we can use the commands “service” or “systemctl”
  - sudo systemctl start ssh

HOW CAN I SEE MY MACHINE'S SERVICES?

- Service managers:



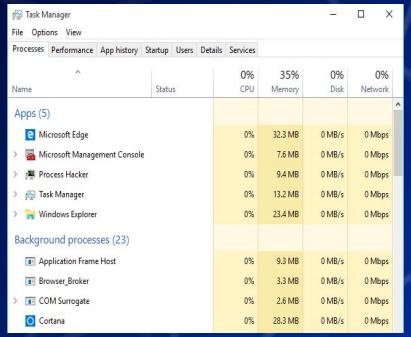
Windows:
Task Manager
Services Tab



Linux:
systemctl
list-unit-files

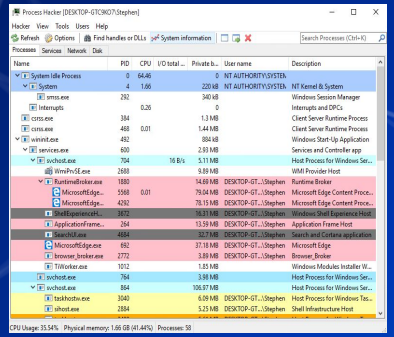
HOW CAN I SEE MY MACHINE'S PROCESSES?

- Process managers:



Task Manager Process Tab

Windows Built-in



Process Hacker

Windows Freeware

```
root      8603  0.0  0.0      0     0 ?        S    17:58   0:00 [kworker/6:1]
root      8625  0.0  0.0 165180  6212 ?        Ss   17:58   0:00 sshd: vzheng8 [
vzheng8   8637  0.0  0.0 165180  2700 ?        S    17:58   0:00 sshd: vzheng8@n
vzheng8   8638  0.0  0.0 121368  1604 ?        Ss   17:58   0:00 tcsh -c /usr/li
vzheng8   8654  0.0  0.0  74292  2920 ?        S    17:58   0:00 /usr/libexec/op
root      8858  0.0  0.0      0     0 ?        S    18:01   0:00 [kworker/4:0]
root      8970  0.0  0.0 163068  5784 ?        Ss   Sep30   0:00 sshd: regan [pr
regan     8975  0.0  0.0 163068  2628 ?        S    Sep30   0:00 sshd: regan@not
regan     8976  0.0  0.0 121368  1608 ?        Ss   Sep30   0:00 tcsh -c /usr/li
regan     8994  0.0  0.0  74292  3040 ?        S    Sep30   0:00 /usr/libexec/op
root      9809  0.0  0.0      0     0 ?        S    Oct01   0:00 [kworker/13:0]
anarghya  9972  0.0  0.0 107952   408 ?        S    18:18   0:00 sleep 180
root     10013  0.5  0.0 163080  5984 ?        Ss   18:19   0:00 sshd: sjames5 [
sjames5  10023  0.0  0.0 163080  2476 ?        R    18:19   0:00 sshd: sjames5@p
sjames5  10024  0.1  0.0 121628  2104 pts/2    Ss   18:19   0:00 -tcsh
root     10069  0.0  0.0 107952   356 ?        S    18:19   0:00 sleep 60
root     10097  0.0  0.0      0     0 ?        S    18:20   0:00 [kworker/2:2]
sjames5  10125  0.0  0.0 157452  1924 pts/2    R+   18:20   0:00 ps aux
root     11130  0.0  0.0 163068  5800 ?        Ss   Oct01   0:00 sshd: regan [pr
regan    11140  0.0  0.0 163068  2852 ?        S    Oct01   0:00 sshd: regan@pts
regan    11141  0.0  0.0 121624  2116 pts/1    Ss+  Oct01   0:00 -tcsh
root     11643  0.0  0.0      0     0 ?        S<   Sep06   1:31 [kworker/15:2H]
```

$ ps

Linux Built-in

```
top - 18:19:56 up 32 days, 18:07,  6 users,  load average: 0.00, 0.01, 0.05
Tasks: 275 total,   1 running, 272 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 32932400 total, 26738652 free,   456824 used,  5736924 buff/cache
KiB Swap: 32767996 total, 31865596 free,   902400 used. 31371832 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
10057 sjames5   20   0  164236   2468   1624 R   0.7  0.0   0:00.16 top
 3058 anarghya  20   0 2093048  51240  16120 S   0.3  0.2   0:05.80 node
    1 root      20   0  194816   5952   2724 S   0.0  0.0  20:11.37 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:02.54 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:02.43 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:+
    6 root      20   0       0      0      0 S   0.0  0.0   1:09.37 kworker/u6+
    8 root      rt   0       0      0      0 S   0.0  0.0   0:00.93 migration/0
    9 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
   10 root      20   0       0      0      0 S   0.0  0.0   9:21.24 rcu_sched
   11 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 lru-add-dr+
   12 root      rt   0       0      0      0 S   0.0  0.0   0:30.28 watchdog/0
   13 root      rt   0       0      0      0 S   0.0  0.0   0:07.69 watchdog/1
   14 root      rt   0       0      0      0 S   0.0  0.0   0:00.45 migration/1
   15 root      20   0       0      0      0 S   0.0  0.0   0:00.84 ksoftirqd/1
   17 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:+
   19 root      rt   0       0      0      0 S   0.0  0.0   0:07.20 watchdog/2
```

$ top

Linux Built-in

FINDING HARDER TO SEE SERVICES:

- Scan your network/hosts
  - Red and Blue team tactic
- Network/host scans can expose ports that are open/closed/filtered
- Open ports show which services might be running
  - Tools like `nmap` provide further detail on which specific services (including versions) are installed

DEEPEST SERVICES DIVE:

- Further means exist to show exactly which services are running when!
- Configuration files
  - Databases, remote access, web, file transfer
- Logs
  - All of the above AND
  - File system journals, security logs, system logs, etc.
  - Where:

IMPLEMENTATION 3: NMAP activity

- Install nmap on any linux box on your LAN
  - Read man page on how nmap works
  - Run a port scan on the entire LAN subnet
  - Save this scan to a file
  - nmap [options] [ip address/subnet]

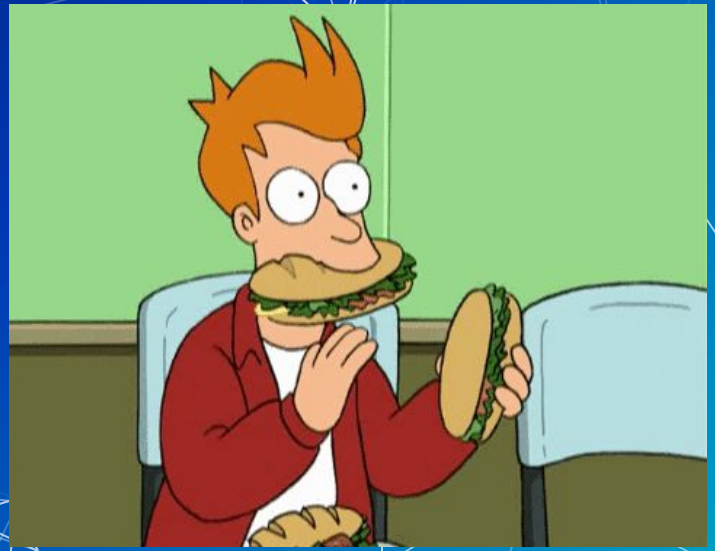
YOUR HOMEWORK

- Set up Mediawiki on WEB
- Set up MariaDB on Rocky
  - Web will connect to this remotely and retrieve database information
- TIP: SCP is used to transfer files, it should be used to take a local file and put it on a remote system

WHAT QUESTIONS DO YOU HAVE?

ROCK ‘N’ ROLL

@XPHILFOX



ASSERT DOMINANCE

@luke