Security Workshop Handon Server Config

Physical Network

The handon server will connect to the Internet by some means (like via the Hotel network, etc). The handson network will be established behind the handson server and the handson server will act as a border gateway for that segment.

Since the users need to directly connect to the NATed segment, we will prepare a WiFi bridge using a USB network adapter.

The NATed segment will be managed by `lxd`. To bridge the WiFi adapter and the NATed segment, please follow the network configuration part of the `lxd` described later.

IP Addresses

The followings are the IP address assignment for this handson system. The subnetmasks are 10.0.0.0/16 for IPv4 and fd00:2497:1::/64 for IPv6.

Node	IPv4	IPv6
ns01	10.0.0.1	fd00:2497:1::1
...	...	...
nsXX	10.0.0.X	fd00:2497:1::X
HTTP proxy	10.0.255.1	n/a
Fakeroot DNS	10.0.255.10	fd00:2497:1::255:10
Workshop DNS	10.0.255.11	fd00:2497:1::255:11
Full Resover	10.0.255.12	fd00:2497:1::255:12
Router Adv	10.0.255.21	fd00:2497:1::255:21

Handson System Preparation

Install Linux Contianer Package

```
sudo apt install lxd lxd-client
```

Initialize `lxd`

```
sudo lxd init
```

Update `lxdbr0`

Update the `lxdbr0` bridge configuration.

```
lxc network edit lxdbr0
```

The updated configuration section looks like something below.

```
config:
  bridge.external_interfaces: YOUR_USB_ETHER_IFNAME
  ipv4.address: 10.0.255.1/16
  ipv4.dhcp: "false"
  ipv4.routing: "false"
  ipv6.address: none
  ipv6.routing: "false"
```

We use a WiFi adapter to accept handson participants, and the adapter is bridged to the `lxdbr0` interface using a USB Ethernet adapter. You need to configure the `bridge.external_interfaces` parameter propery. The interface name of the USB adapter (or whatever) is something like `enx00e04c030ffd`.

DHCP/DHCPv6 Server

Since we do not use the DHCP/DHCPv6 function provided by the `lxc` system, we need to prepare our own.

Install the `isc-dhcp-server`.

```
sudo apt install isc-dhcp-server
```

Configure the DHCP address range for the `lxdbr0` interfaces defined in `/etc/dhcp/dhcpd.conf`.

```
subnet 10.0.0.0 netmask 255.255.0.0 {
  max-lease-time 1800;
  range 10.0.1.0 10.0.254.255;
  option routers 10.0.255.1;
  option domain-name "workshop";
  option domain-name-servers 10.0.255.12;
}
```

Configure the DHCP6 configuration file to advertise IPv6 DNS server address in `/etc/dhcp/dhcpd6.conf`.

```
subnet6 fd00:2497:1::/64 {
        option dhcp6.name-servers fd00:2497:1::255:12;
        option dhcp6.domain-search "workshop";
        option dhcp6.info-refresh-time 1800;
}
```

Edit `/etc/default/isc-dhcp-server` file.

```
INTERFACESv4="lxdbr0"
INTERFACESv6="lxdbr0"
```

Make sure that DHCP/DHCPv6 services are active and enabled.

```
systemctl status isc-dhcp-server
systemctl status isc-dhcp-server6
```

HTTP Proxy

In the handson network, we build a copy of the Internet environment. Since we provide our own DNS tree, it is not possible for the nodes inside the handson network to reach the web contents outside. We provide a proxy service especially for the software installation process for container instances.

```
sudo apt install squid
```

Configure the parameters defined in `/etc/squid/squid.conf`.

```
acl localnet src 10.0.0.0/16

...

http_access allow localnet

...

http_port 10.0.255.1:3128
```

Download Container Images

```
lxc image copy ubuntu:18.04 local: --alias ubuntu18.04
```

Create Custom Images

```
lxc launch ubuntu18.04
```

Configure the system and save the image. See the other sections for the radvd server and name servers.

```
lxc publish SOURCE_CONTAINER_NAME --alias IMAGE_NAME
```

Scripts

Initialize servers

Initialize (Delete and Create) the handson infrastructure servers.

```
#!/bin/sh
#
# secws_init_servers.sh

# radvd server
lxc delete -f radvd
lxc init radvd radvd

# fake root dns server
lxc delete -f dns-fakeroot
lxc init dns-fakeroot dns-fakeroot

# workshop. dns server
lxc delete -f dns-workshop
lxc init dns-workshop dns-workshop
```

Initialize user conainers

Initialize (Delete and Create) the handson user instances.

```
#!/bin/sh
#
# secws_init.sh

NUSERS=3

for i in `seq 1 ${NUSERS}`
do
        ID=`printf %02d ${i}`
        lxc delete -f ns${ID}
        lxc init secws ns${ID}
        cat > /tmp/hosts <<EOF
127.0.0.1 localhost ns${ID}

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
EOF
        cat > /tmp/50-cloud-init.yaml <<EOF
# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    version: 2
    ethernets:
        eth0:
            dhcp4: false
            dhcp6: false
            addresses:
                - 10.0.0.${i}/16
                - fd00:2497:1::${i}/64
            nameservers:
                addresses:
                    - 10.0.255.11
                    - fd00:2497:1::255:11
            accept-ra: false
EOF
        lxc file push /tmp/50-cloud-init.yaml ns${ID}//etc/netplan/
        rm -f /tmp/50-cloud-init.yaml
done
```

Start hands-on containers

Starting the handson instances. You need to be careful when you startup just after host is rebooted, since some of the services depend on other services.

In APRICOT 2019, the `lxd` service must be restarted by `systemctl restart lxd` otherwise, the service didn't work properly. I didn't track down the source of the issue...

Also, the host side of DHCPv4/squid servers depend on the availability of the `lxdbr0` IPv4 network. The host side of DHCPv6 service depends on the `lxdbr0` IPv6 address which will be assigned by the radvd container server. So after the initial boot, please make sure if you have following services are working.

- USB Ethernet (to wireless bridge) is up
- squid is up and running (`systemctl status squid`)
- DHCPv4 is up and running (`systemctl status isc-dhcp-server`)
- DHCPv6 is up and running (`systemctl status isc-dhcp-server6`)

```
#!/bin/sh
#
# secws_start.sh

for i in `lxc list -c n --format csv`
do
        echo Starting ${i}
        lxc start ${i}
done
```

Stop hands-on containers

Stopping the handson instances (with keeping the current image status).

```
#!/bin/sh
#
# secws_stop.sh

for i in `lxc list -c n --format csv`
do
        echo Stopping ${i}
        lxc stop ${i}
done
```

Setup Router Adv Server

Create a new container from the base image and login to the node.

```
lxc launch ubuntu18.04 radvd
lxc exec radvd bash
```

Install `radvd`

```
apt install radvd
```

Configure the `/etc/radvd.conf` file. `AdvOtherConfigFlag` is required to provide IPv6 DNS server address via DHCPv6.

```
interface eth0 {
        AdvSendAdvert on;
        AdvOtherConfigFlag on;
        prefix fd00:2497:1::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
        };
        RDNSS fd00:2497:1::255:12
        {
                AdvRDNSSLifetime 1800;
        };
};
```

Make sure the `radvd` program starts on boot.

```
systemctl enable radvd
```

Address Configuration

Disable DHCP/DHCPv6 operation by creating the file `/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`. The contents will be as below.

```
network: {config: disabled}
```

Create a new netplan configurtion file at `/etc/netplan/50-cloud-init.yaml`.

```
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.0.255.21/16
                - fd00:2497:1::255:21/64
            accept-ra: false
            nameservers:
                addresses:
                    - 10.0.255.11
                    - fd00:2497:1::255:11
```

Enable `tcpdump`

For trouble shooting, we sometimes need to monitor packets. To enable `tcpdump` command, update the `apparmer` configuration.

```
cd /etc/apparmor.d/disable
ln -s /etc/apparmor.d/usr.sbin/tcpdump
```

Save the Image

Exit the container and publish the image.

```
lxc stop radvd
lxc publish radvd --alias radvd
```

Setup the Fake Root Name Server

Create a new container from the base image and login to the node.

```
lxc launch ubuntu18.04 dns-fakeroot
lxc exec dns-fakeroot bash
```

Install `bind9`

Install the `bind9` package and configure for the fake root name server.

```
apt install bind9
```

Generate Kyes

```
cd /etc/bind
mkdir keys.fakeroot
cd keys.fakeroot
# Generate a key signing key (KSK)
dnssec-keygen -f KSK -3 -a ECDSAP256SHA256 -r /dev/urandom .
# Generate a zone signing key (ZSK)
dnssec-keygen -3 -a ECDSAP256SHA256 -r /dev/urandom .
```

Create the Fake Root Zone File

```
; /etc/bind/db.fakeroot
$TTL    1800
@       IN      SOA     a.fakeroot-servers.net. root.a.fakeroot-servers.net. (
                      2      ; Serial
                 604800      ; Refresh
                  86400      ; Retry
                2419200      ; Expire
                    300 )    ; Negative Cache TTL
;
@                       IN NS   a.fakeroot-servers.net.
a.fakeroot-servers.net IN A    10.0.255.10
a.fakeroot-servers.net IN AAAA fd00:2497:1::255:10
```

Sign the Root Zone File

```
cd /etc/bind
dnssec-signzone -S -K /etc/bind/keys.fakeroot/ -a -r /dev/urandom -o . db.fakeroot
```

Update `named.conf.default-zones`

The configuration file `/etc/bind/named.conf.default-zones` must have the root zone information as a master server.

```
zone "." {
        type master;
        file "/etc/bind/db.fakeroot.signed";
};
```

Restart Name Server

```
systemctl stop bind9
systemctl start bind9
```

Address Configuration

Disable DHCP/DHCPv6 operation by creating the file `/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`. The contents will be as below.

```
network: {config: disabled}
```

Create a new netplan configurtion file at `/etc/netplan/50-cloud-init.yaml`.

```
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.0.255.10/16
                - fd00:2497:1::255:10/64
            accept-ra: false
```

Enable `tcpdump`

For trouble shooting, we sometimes need to monitor packets. To enable `tcpdump` command, update the `apparmer` configuration.

```
cd /etc/apparmor.d/disable
ln -s /etc/apparmor.d/usr.sbin/tcpdump
```

Save the Image

Exit the container and publish the image.

```
lxc stop dns-fakeroot
lxc publish dns-fakeroot --alias dns-fakeroot
```

Setup the `workshop.` Name Server

Create a new container from the base image and login to the node.

```
lxc launch ubuntu18.04 dns-workshop
lxc exec dns-workshop bash
```

Install `bind9`

Install the `bind9` package and configure for the `workshop.` domain name server.

```
apt install bind9
```

Disable Recursion

To serve as an authoritative only resolver, update the `options` section of the `/etc/bind/named.conf.options` file.

```
options {
...
        recursion no;
};
```

Generate Kyes

```
cd /etc/bind
mkdir keys.workshop
cd keys.workshop
# Generate a key signing key (KSK)
dnssec-keygen -f KSK -3 -a ECDSAP256SHA256 -r /dev/urandom workshop
# Generate a zone signing key (ZSK)
dnssec-keygen -3 -a ECDSAP256SHA256 -r /dev/urandom workshop
```

Create the `workshop.` Zone File

```
; /etc/bind/db.workshop
$TTL    300
@            IN  SOA     ns.sec. root.ns.sec. (
                      2      ; Serial
                 604800      ; Refresh
                  86400      ; Retry
                2419200      ; Expire
                    300 )    ; Negative Cache TTL
;
@            IN  NS      ns.workshop.
ns.workshop.    IN  A       10.0.255.11
ns.workshop.    IN  AAAA    fd00:2497:1::255:11
```

Sign the `workshop.` Zone File

```
cd /etc/bind
dnssec-signzone -S -K /etc/bind/keys.workshop/ -a -r /dev/urandom -o workshop db.w
orkshop
```

Setup the DS record

Setup glue records of `workshop.` domain in the fake root name server. The required information is the DS records generated in the above signing process and NS/A records.

The DS record information is in the file `/etc/bind/dsset-workshop.` file of the `dns-workshop` container.

Go to the `dns-fakeroot` container, and append the following information to the `db.fakeroot` file.

```
workshop.                IN      NS      ns.workshop.
ns.workshop.             IN      A       10.0.255.11
ns.workshop.             IN      AAAA    fd00:2497:1::255:11
workshop.                IN DS 65048 13 1 C138A8EAB1DC564788B03BFA09A1EE6B278B5A83
workshop.                IN DS 65048 13 2 C2DFC2B9D009B21345FB951A76A06779EC2E95445
D709FA9BAD0A7D0 827678EC
```

Sign the Root Zone

Since we have updated the root zone, sign the root zone again in the root name server.

```
dnssec-signzone -S -K /etc/bind/keys.fakeroot/ -g -a -r /dev/urandom -o . db.faker
oot
```

Then restart the bind software on the fake root name server container.

Replace the Root Hint

Go back to the `dns-workshop` container.

Since we are using our own (fake) root name server, we need to update the root hint file `/etc/bind/db.root`.

```
.                          1800      NS    a.fakeroot-servers.net.
a.fakeroot-servers.net.    1800      A     10.0.255.10
a.fakeroot-servers.net.    1800      AAAA  fd00:2497:1::255:10
```

Replace the Trust Anchor

We also need to replace the trust anchor file `/etc/bind/bind.keys`.

```
# /etc/bind/bind.keys
managed-keys {
    . initial-key 257 3 13 "I9jR4QIl/xSDyXgByM+Y51NNnHGLsVEUQy+z
                            DjaMLSLSJTT8icKPmTVXmyu7md85gPckbqMQ
                            CjvAZ5tXfpdILA==";
};
```

Replace the key information with the KSK value of the fake root name server generated previously.

Update Configration Files

Append the information for `workshop.` zone to `/etc/bind/named.conf.local`.

```
...
zone "workshop" {
        type master;
        file "/etc/bind/db.workshop.signed";
};
```

Address Configuration

Disable DHCP/DHCPv6 operation by creating the file `/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`. The contents will be as below.

```
network: {config: disabled}
```

Create a new netplan configurtion file at `/etc/netplan/50-cloud-init.yaml`.

```
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.0.255.11/16
                - fd00:2497:1::255:11/64
            accept-ra: false
```

Enable `tcpdump`

For trouble shooting, we sometimes need to monitor packets. To enable `tcpdump` command, update the `apparmer` configuration.

```
cd /etc/apparmor.d/disable
ln -s /etc/apparmor.d/usr.sbin/tcpdump
```

Save the Image

Exit the container and publish the image.

```
lxc stop dns-workshop
lxc publish dns-workshop --alias dns-workshop
```

Setup the cache Name Server

Create a new container from the base image and login to the node.

```
lxc launch ubuntu18.04 dns-cache
lxc exec dns-cache bash
```

Install `bind9`

Install the `bind9` package.

```
apt install bind9
```

Configure ACL for recursive quiery

To serve as a full resolver from clients in the handson segment, update the `options` section of the `/etc/bind/named.conf.options` file.

```
options {
...
        recursion yes;
        allow-recursion {
                localhost;
                localnets;
        };
};
```

Replace the Root Hint

Since we are using our own (fake) root name server, we need to update the root hint file `/etc/bind/db.root`.

```
.                         1800      NS    a.fakeroot-servers.net.
a.fakeroot-servers.net.   1800      A     10.0.255.10
a.fakeroot-servers.net.   1800      AAAA  fd00:2497:1::255:10
```

Replace the Trust Anchor

We also need to replace the trust anchor file `/etc/bind/bind.keys`.

```
# /etc/bind/bind.keys
managed-keys {
    . initial-key 257 3 13 "I9jR4QIl/xSDyXgByM+Y51NNnHGLsVEUQy+z
                            DjaMLSLSJTT8icKPmTVXmyu7md85gPckbqMQ
                            CjvAZ5tXfpdILA==";
};
```

Replace the key information with the KSK value of the fake root name server generated previously.

Address Configuration

Disable DHCP/DHCPv6 operation by creating the file `/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`. The contents will be as below.

```
network: {config: disabled}
```

Create a new netplan configurtion file at `/etc/netplan/50-cloud-init.yaml`.

```
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.0.255.12/16
                - fd00:2497:1::255:12/64
            accept-ra: false
            nameservers:
                addresses:
                    - 10.0.255.12
                    - fd00:2497:1::255:12
```

Resolver Configuration

To bypass systemd's resolver, replace the `/etc/resolv.conf` to point the nameservers directly.

```
cd /etc
rm -f resolv.conf
ln -s /run/systemd/resolve/resolv.conf
```

Enable `tcpdump`

For trouble shooting, we sometimes need to monitor packets. To enable `tcpdump` command, update the `apparmer` configuration.

```
cd /etc/apparmor.d/disable
ln -s /etc/apparmor.d/usr.sbin/tcpdump
```

Save the Image

Exit the container and publish the image.

```
lxc stop dns-cache
lxc publish dns-cache --alias dns-cache
```

Setup the Handson Virtual Server

Create a new container from the base image and login to the node.

```
lxc launch ubuntu18.04 secws
lxc exec secws bash
```

Setup ssh

To allow users to login to this server with password for the first time, update the `/etc/ssh/sshd_config`.

```
PasswordAuthentication yes
```

Create `workshop` account

Create the user account for handson.

```
adduser workshop
```

The initial UNIX password should be `iij/2497`.

Sudo setup

Add the `workshop` user to the `sudo` group.

```
vigr
```

and update like below.

```
...
sudo:x:27:ubuntu,workshop
...
```

Resolver Configuration

To bypass systemd's resolver, replace the `/etc/resolv.conf` to point the nameservers directly.

```
cd /etc
rm -f resolv.conf
ln -s /run/systemd/resolve/resolv.conf
```

APT HTTP proxy

To enable workshop users to update or install software from the workshop networ, the APT HTTP proxy must be configures. Create `/etc/apt/apt.conf.d/99proxy.conf`.

```
Acquire::http::Proxy "http://10.0.255.1:3128/";
Acquire::https::Proxy "http://10.0.255.1:3128/";
```

Enable `tcpdump`

For trouble shooting, we sometimes need to monitor packets. To enable `tcpdump` command, update the `apparmer` configuration.

```
cd /etc/apparmor.d/disable
ln -s /etc/apparmor.d/usr.sbin/tcpdump
```

Save the Image

Exit the container and publish the image.

```
lxc stop secws
lxc publish secws --alias secws
```