User Manual

Environment Setup

The following document contains a copy of the details outlined in the section 2.5.3 of the Major Project report document.

The environment for performing development and testing was created using Vagrant, a tool for defining software infrastructure as code. The changes to the source code are automatically synchronized between the VMs and the host.

```
Vagrant.configure("2") do |config|
  config.vm.define "nyako" do |server|
    server.vm.box = "fedora/34-cloud-base"
    server.vm.network "private_network", ip: "192.168.56.11"
    server.vm.hostname = "nyako"
    server.vm.define "nyako"
    server.vm.provision :shell, path: "setup_nyako.sh"
    server.vm.synced_folder "nyako", "/home/vagrant/nyako"
    server.vm.provider "virtualbox" do |vb|
      vb.memory = "2048"
      vb.cpus = "2"
    end
  end

  config.vm.define "nyatta" do |client|
    client.vm.box = "fedora/34-cloud-base"
    client.vm.network "private_network", ip: "192.168.56.12"
    client.vm.hostname = "nyatta"
    client.vm.define "nyatta"
    client.vm.provision :shell, path: "setup_nyatta.sh"
    client.vm.synced_folder "nyatta", "/home/vagrant/nyatta"
    client.vm.provider "virtualbox" do |vb|
      vb.memory = "2048"
      vb.cpus = "2"
    end
  end
end
```

Figure 1: Vagrantfile.

The environment consists of two virtual machines with the following characteristics:

- Target Host
  - Operating System: Fedora 34
  - IP address: 192.168.56.11
  - Hostame: “nyako”
- Attacker Host:
  - Operating System: Fedora 34
  - IP address: 192.168.56.12

- Hostame: “nyatta”

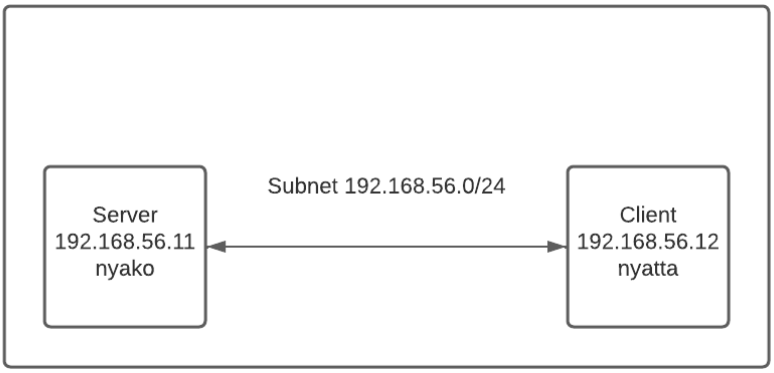

Figure 2: development environment setup.

The virtual machines are defined inside the Vagrantfile located at the root directory. The packages necessary for development and testing are defined inside setup_nyako.sh and setup_nyatta.sh bash scripts.

The environment can be created using the *vagrant up* command. After the commands execution the VMs are accessible via *ssh <hostname>* command.

```
[maksymc@maksym-hpspectrex360convertible13aw0xxx Source]$ vagrant up
Bringing machine 'nyako' up with 'virtualbox' provider...
Bringing machine 'nyatta' up with 'virtualbox' provider...
==> nyako: Importing base box 'fedora/34-cloud-base'...
==> nyako: Matching MAC address for NAT networking...
==> nyako: Checking if box 'fedora/34-cloud-base' version '34.20210423.0' is up to date...
==> nyako: Setting the name of the VM: Source_nyako_1648339159510_92889
==> nyako: Clearing any previously set network interfaces...
==> nyako: Preparing network interfaces based on configuration...
    nyako: Adapter 1: nat
    nyako: Adapter 2: hostonly
==> nyako: Forwarding ports...
    nyako: 22 (guest) => 2222 (host) (adapter 1)
==> nyako: Running 'pre-boot' VM customizations...
==> nyako: Booting VM...
==> nyako: Waiting for machine to boot. This may take a few minutes...
```

Figure 3: development environment creation.

Compilation

Nyako and Nyatta can be compiled using the available Makefiles.

Nyatta

To compile Nyatta run: *make nyatta*.

```
[root@nyatta nyatta]# make nyatta
gcc -Wall -o nyatta src/logger.c src/utils.c src/crypto.c src/message
.c src/network.c src/listener.c src/nyatta.c -lpcap -lsodium -pthread
 -lcurl -lnet
```

Figure 4: Nyatta compilation.

Nyako

To compile Nyako run: *make nyako*.

```
[root@nyako nyako]# make nyako
clang -g -O2 -Wall -target bpf -c src/nyako_kern.c -o nyako_kern.o
clang -g -O2 -Wall -target bpf -c src/no_trace_kern.c -o no_trace_kern.o
bpftool gen skeleton no_trace_kern.o > src/no_trace_skeleton.h
clang -g -O2 -Wall -target bpf -c src/pidhide_kern.c -o pidhide_kern.o
bpftool gen skeleton pidhide_kern.o > src/pidhide_skeleton.h
clang -g -O2 -Wall -o nyako.o src/utils.c src/logger.c src/crypto.c src/message.c sr
c/network.c src/bpf_helpers.c src/no_trace.c src/pidhide.c src/nyako.c -lsodium -lcu
rl -lbpf -pthread
```

Figure 5 Nyako compilation.

Usage Instructions

Nyato and Nyatta do not accept any command line arguments, the rootkit configuration can be performed by updating corresponding config.h files. The configuration options are self explanatory. The cheatsheet.txt file contains commonly used helper commands.

```
[root@nyako nyako]# ./nyako.o
hiding PID 3662 ...
```

Figure 6: Nyako execution.

Remote command execution can be performed by entering a linux command. Rootkit specific commands are outlined in the table below.

Command	Description
invoke	Send a message with the command type TYPE_INVOKE.
suspend	Send a message with the command type TYPE_SUSPEND.
block_trace	Send a message with the command type TYPE_BLOCK_TRACE.
unblock_trace	Send a message with the command type TYPE_UNBLOCK_TRACE.
terminate	Send a message with the command type TYPE_TERMINATE.

Table 1: rootkit specific commands.

```
[root@nyatta nyatta]# ./nyatta
INFO: created data loop with filter: src host 192.168.56.11 and port 80
terminate
```

Figure 7: Nyatta execution.