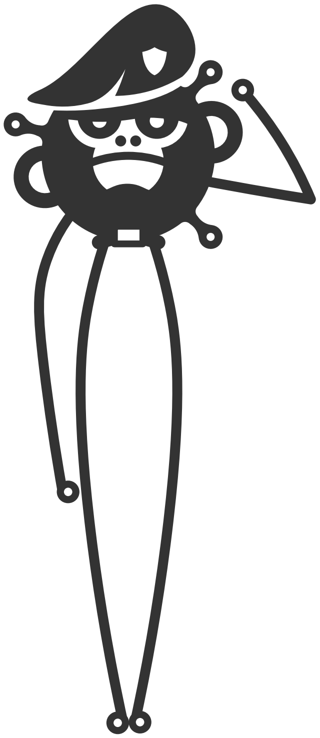


MONKEYZOO

GCP testing network for Infection Monkey

PURPOSE

This document describes each machine in Infection Monkey’s private test network and is intended for developers only.

Guardicore^{TM}

Warning!

This project builds an intentionally vulnerable network. Make sure not to add production servers to the same network and leave it closed to the public.

Introduction:

MonkeyZoo is a Google Cloud Platform network deployed with terraform. Terraform scripts allows you to quickly setup a network that’s full of vulnerable machines to regression test monkey’s exploiters, evaluate scanning times in a real-world scenario and many more.

Getting started:

Requirements:

1. Have terraform installed.
2. Have a Google Cloud Platform account (upgraded if you want to test whole network at once).

To deploy:

1. Crete a service account for your project named “you_name-monkeyZoo-user” and download its **Service account key**. Select JSON format.
2. Get these permissions in monkeyZoo project for your service account:
    a. **Compute Engine -> Compute image user**
3. Change configurations located in the ../monkey/envs/monkey_zoo/terraform/config.tf file (don’t forget to link to your service account key file):

```
provider "google" {
        project = "project-28054666"
        region  = "europe-west3"
        zone    = "europe-west3-b"
        credentials = "${file("project-92050661-9dae6c5a02fc.json")}"
}
service_account_email="test@project-925243.iam.gserviceaccount.com"
```

4. Run `terraform init`

To deploy the network run:

```
terraform plan          (review the changes it will make on GCP)
terraform apply         (creates 2 networks for machines)
terraform apply         (adds machines to these networks)
```

Using islands:

How to get into the islands:

island-linux-250: SSH from GCP

island-windows-251: In GCP/VM instances page click on island-windows-251. Set password for your account and then RDP into the island.

These are most common steps on monkey islands:

island-linux-250:

To run monkey island:

`sudo /usr/run_island.sh`

To run monkey:

`sudo /usr/run_monkey.sh`

To update repository:

1. `git pull /usr/infection_monkey/`

Update all requirements using deployment script:

1. `cd /usr/infection_monkey/deployment_scripts`
2. `./deploy_linux.sh "/usr/infection_monkey" "develop"`

island-windows-251:

To run monkey island:

Execute C:\run_monkey_island.bat as administrator

To run monkey:

Execute C:\run_monkey.bat as administrator

To update repository:

1. Open cmd as an administrator
2. `cd C:\infection_monkey`
3. `git pull` (updates develop branch)

Update all requirements using deployment script:

1.` cd C:\infection_monkey\deployment_scripts`
2. ` ./run_script.bat "C:\infection_monkey" "develop"`

Running tests:

Once you start monkey island you can import test configurations from ../monkey/envs/configs.

fullTest.conf is a good place to start, because it covers all machines.

Machines' legend:

"Machines" paragraph describes each network machine one by one.

Background colours meaning:

Red: machine is exploited using credentials from configuration (brute-force attack).

Blue: machine is exploited trough a vulnerability (no credentials needed).

Green: machine is secure.

Grey: machine is not implemented/doesn't work yet.

Machines:

Nr. 2 Hadoop **(10.2.2.2)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	JDK, Hadoop 2.9.1
Default server’s port:	8020
Server’s config:	Single node cluster
Scan results:	Machine exploited using Hadoop exploiter
Notes:	

Nr. 3 Hadoop **(10.2.2.3)**	
OS:	**Windows 10 x64**
Software:	JDK, Hadoop 2.9.1
Default server’s port:	8020
Server’s config:	Single node cluster
Scan results:	Machine exploited using Hadoop exploiter
Notes:	

Nr. 4 Elastic **(10.2.2.4)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	JDK, Elastic 1.4.2
Default server's port:	9200
Server's config:	Default
Scan results:	Machine exploited using Elastic exploiter
Notes:	Quick tutorial on how to add entries (was useful when setting up).

Nr. 5 Elastic **(10.2.2.5)**	
OS:	**Windows 10 x64**
Software:	JDK, Elastic 1.4.2
Default server's port:	9200
Server's config:	Default
Scan results:	Machine exploited using Elastic exploiter
Notes:	Quick tutorial on how to add entries (was useful when setting up).

Nr. 6 Sambacry

(10.2.2.6)

OS:	**Ubuntu 16.04.05 x64**
Software:	Samba > 3.5.0 and < 4.6.4, 4.5.10 and 4.4.14
Default server's port:	-
Root password:	;^TK`9XN_x^
Server's config:	
Scan results:	Machine exploited using Sambacry exploiter
Notes:	

Nr. 7 Sambacry

(10.2.2.7)

OS:	**Ubuntu 16.04.05 x32**
Software:	Samba > 3.5.0 and < 4.6.4, 4.5.10 and 4.4.14
Default server's port:	-
Root password:	*.&A7/W}Rc$
Server's config:	
Scan results:	Machine exploited using Sambacry exploiter
Notes:	

Nr. 8 Shellshock **(10.2.2.8)**	
OS:	**Ubuntu 12.04 LTS x64**
Software:	Apache2, bash 4.2.
Default server’s port:	80
Scan results:	Machine exploited using Shellshock exploiter
Notes:	Vulnerable app is under /cgi-bin/test.cgi

Nr. 9 Tunneling M1 **(10.2.2.9, 10.2.1.9)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	OpenSSL
Default service’s port:	22
Root password:	`))jU7L(w}
Server’s config:	Default
Notes:	

Nr. 10 Tunneling M2 **(10.2.1.10)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	OpenSSL
Default service’s port:	22
Root password:	3Q=(Ge(+&w]*
Server’s config:	Default
Notes:	Accessible only trough Nr.9

Nr. 11 SSH key steal.

(10.2.2.11)

OS:	**Ubuntu 16.04.05 x64**
Software:	OpenSSL
Default connection port:	22
Root password:	^NgDvY59~8
Server's config:	SSH keys to connect to NR. 11
Notes:	

Nr. 12 SSH key steal.

(10.2.2.12)

OS:	**Ubuntu 16.04.05 x64**
Software:	OpenSSL
Default connection port:	22
Root password:	u?Sj5@6(-C
Server's config:	SSH configured to allow connection from NR.10
Notes:	Don't add this machine's credentials to exploit configuration.

Nr. 13 RDP grinder **(10.2.2.13)**	
OS:	**Windows 10 x64**
Software:	-
Default connection port:	3389
Root password:	2}p}aR]&=M
Scan results:	Machine exploited using RDP grinder
Server’s config:	Remote desktop enabled Admin user’s credentials: m0nk3y, 2}p}aR]&=M
Notes:	

Nr. 14 Mimikatz

(10.2.2.14)

OS:	**Windows 10 x64**
Software:	-
Admin password:	Ivrrw5zEzs
Server's config:	Has cashed mimikatz-15 RDP credentials SMB turned on
Notes:	

Nr. 15 Mimikatz

(10.2.2.15)

OS:	**Windows 10 x64**
Software:	-
Admin password:	pAJfG56JX><
Server's config:	It's credentials are cashed at mimikatz-14 SMB turned on
Notes:	If you change this machine's IP it won't get exploited.

Nr. 16 MsSQL

(10.2.2.16)

OS:	**Windows 10 x64**
Software:	MSSQL Server
Default service port:	1433
Server’s config:	xp_cmdshell feature enabled in MSSQL server Server’s creds (sa): admin, }8Ys#"
Notes:	Enabled SQL server browser service Enabled remote connections Changed default password

Nr. 17 Upgrader

(10.2.2.17)

OS:	**Windows 10 x64**
Default service port:	445
Root password:	U??7ppG_
Server’s config:	Turn on SMB
Notes:	

Nr. 18 WebLogic **(10.2.2.18)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	JDK, Oracle WebLogic server 12.2.1.2
Default server’s port:	7001
Admin domain credentials:	weblogic : B74Ot0c4
Server’s config:	Default
Notes:	

Nr. 19 WebLogic **(10.2.2.19)**	
OS:	**Windows 10 x64**
Software:	JDK, Oracle WebLogic server 12.2.1.2
Default server’s port:	7001
Admin servers credentials:	weblogic : =ThS2d=m(`B
Server’s config:	Default
Notes:	

Nr. 20 SMB **(10.2.2.20)**	
OS:	**Windows 10 x64**
Software:	-
Default service’s port:	445
Root password:	YbS,<tpS.2av
Server’s config:	SMB turned on
Notes:	

Nr. 21 Scan **(10.2.2.21)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	Apache tomcat 7.0.92
Default server’s port:	8080
Server’s config:	Default
Notes:	Used to scan a machine that has no vulnerabilities (to evaluate scanning speed for e.g.)

Nr. 22 Scan **(10.2.2.22)**	
OS:	**Windows 10 x64**
Software:	Apache tomcat 7.0.92
Default server’s port:	8080
Server’s config:	Default
Notes:	Used to scan a machine that has no vulnerabilities (to evaluate scanning speed for e.g.)

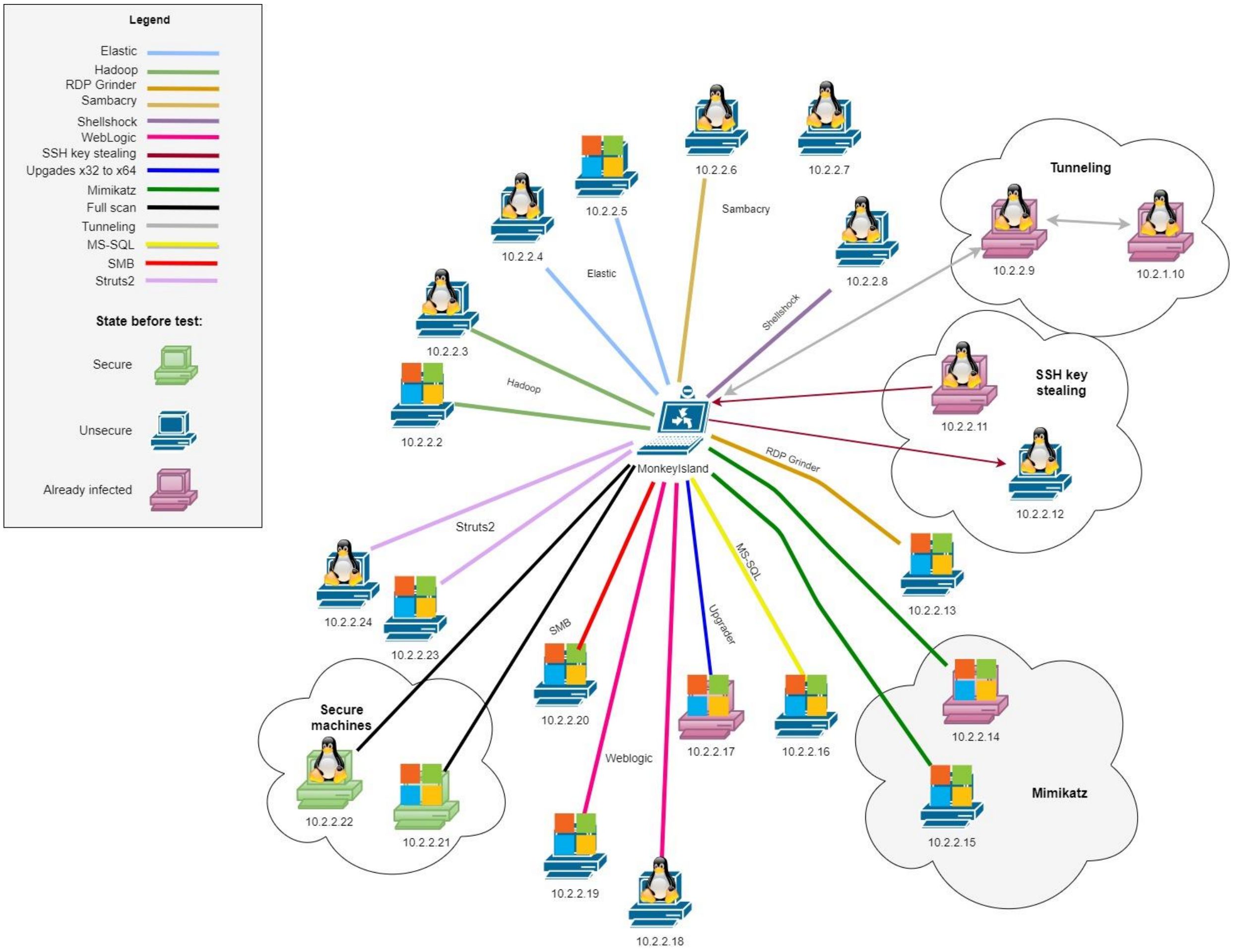
Nr. 23 Struts2 **(10.2.2.23)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	JDK, struts2 2.3.15.1, tomcat 9.0.0.M9
Default server’s port:	8080
Server’s config:	Default
Notes:	

Nr. 24 Struts2 **(10.2.2.24)**	
OS:	**Windows 10 x64**
Software:	JDK, struts2 2.3.15.1, tomcat 9.0.0.M9
Default server’s port:	8080
Server’s config:	Default
Notes:	

Nr. 250 MonkeyIsland **(10.2.2.250)**	
OS:	**Ubuntu 16.04.05 x64**
Software:	MonkeyIsland server, git, mongodb etc.
Default server’s port:	22, 443
Private key passphrase:	-
Notes:	Only accessible trough GCP

Nr. 251 MonkeyIsland **(10.2.2.251)**	
OS:	**Windows Server 2016 x64**
Software:	MonkeyIsland server, git, mongodb etc.
Default server’s port:	3389, 443
Private key passphrase:	-
Notes:	Only accessible trough GCP

Network topography:


Legend
Elastic
Hadoop
RDP Grinder
Sambacry
Shellshock
WebLogic
SSH key stealing
Upgades x32 to x64
Mimikatz
Full scan
Tunneling
MS-SQL
SMB
Struts2
State before test:
Secure
Unsecure
Already infected
10.2.2.6
10.2.2.7
10.2.2.5
10.2.2.4
Sambacry
Elastic
10.2.2.8
Tunneling
10.2.2.9
10.2.1.10
10.2.2.3
Shellshock
Hadoop
10.2.2.2
SSH key stealing
10.2.2.11
MonkeyIsland
RDP Grinder
10.2.2.12
Struts2
10.2.2.24
10.2.2.23
MS-SQL
10.2.2.13
Upgrader
SMB
10.2.2.20
Secure machines
10.2.2.22
10.2.2.21
Weblogic
10.2.2.17
10.2.2.16
10.2.2.14
Mimikatz
10.2.2.15
10.2.2.19
10.2.2.18