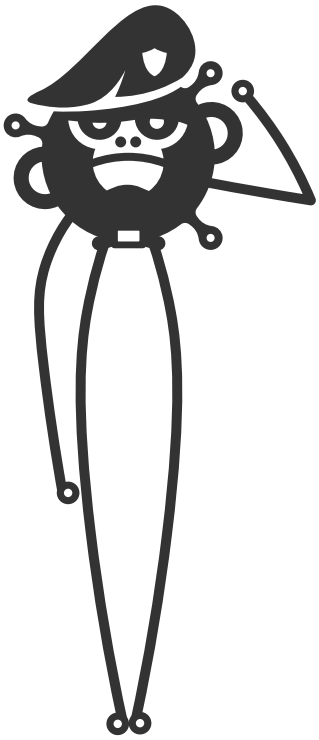

MONKEYZOO

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™

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 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`	(apply those changes)

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.

Accessing machines:

You can access island machines trough rdp/ssh using **m0nk3y** user and password provided in the corresponding machine's documentation.
Other machines are designed in a black-box fashion and should work as soon as they're booted, however it's still possible to access and modify them using GCP API.

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:	Don't forget to add at least a single entry.

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:	Don't forget to add at least a single entry.

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:	-
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:	-
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 Turn on SMB
Scan results:	Machine exploited using SMB Found cashed credentials
Notes:	

Nr. 15 Mimikatz (10.2.2.15)	
OS:	**Windows 10 x64**
Software:	-
Admin password:	pAJfG56JX><
Server's config:	Credentials cashed at mimikatz-14 Turn on SMB
Scan results:	Machine exploited using SMB (creds stolen with mimikatz)
Notes:	Turn on SMB 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:	Add server's credentials to /test/creds before testing Enable SQL server browser service Enable remote connections Change 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:	Turn on SMB
Notes:	Add administrator's password to test/creds

Nr. 21 Scan (10.2.2.21)	
OS:	**Ubuntu 16.04.05 x64**
Software:	-
Default server's port:	-
Server's config:	Default
Notes:	Used to scan a machine with no vulnerabilities (to evaluate scanning speed and etc.)

Nr. 22 Scan (10.2.2.22)	
OS:	**Windows 10 x64**
Software:	-
Default server's port:	-
Server's config:	Default
Notes:	Used to scan a machine with no vulnerabilities (to evaluate scanning speed and etc.)

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:	Defaul
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:	-
Private key passphrase:	05f8jU5ma
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:	-
Private key passphrase:	UXvvuKv5V
Notes:	Only accessible trough GCP

Network topography: