TestingBot.com

OVERVIEW OF SECURITY PROCESSES

November 2017

TestingBot.com is a provider of “browser testing” in the cloud.
We provide an online grid of real browsers, running on Virtual Machines (VMs), on our cloud infrastructure.

Companies connect to our grid to run tests against their website/product on all the different browsers/operating systems we provide.

This document offers an overview of all the processes, technology and features we provide to ensure the security of our customers and their data.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

TestingBot provides a cloud computing platform for functional testing of websites.
The service is located in Belgium, Europe located in a datacenter in Brussels.
Our entire infrastructure runs on our own dedicated hardware. One of the advantages of running our own private cloud is that we are able to set up and optimize the security of our service ourselves. We take the greatest care in making sure the privacy of our users and their data is protected and secure.

This document provides an overview of TestingBot's services together with explanations on how we secure the transfer of data and how we implemented our security policies and procedures.

TESTINGBOT OVERVIEW

Virtual Machines

The TestingBot cloud platform is a grid of Virtual Machines (VMs) which all have a collection of browsers installed. We provide VMs with the most popular operating systems, including Windows, MacOS, Linux and iOS/Android.
For every test that a user runs, our system will allocate a brand new, freshly booted VM. We spin up a pristine VM for every test. Running each test on a new VM is the only way to guarantee that:

- No data from a previous test was still located on the VM
- No other customer could install any kind of program on the VM

Securing shared VMs in a multi-tenant environment is a very difficult job; you can never be sure that the VM used by a previous customer was effectively cleaned. That is why we provide a VM environment where each test runs on a VM that has never before been used and will never be used once your test completes.

A simplified flow of the lifecycle of a TestingBot VM:

1. A VM is booted from scratch, running one of the operating systems we provide
2. A customer wants to run a test on a browser, located on the VM.
3. The test runs on the VM. Any changes happening to the VM are written to a RAM-disk through COW (Copy-On-Write). The base VM is set to read-only.
4. The test finishes. Any assets (screenshots, video, logs) will be sent to AWS S3, if required.
5. The VM is destroyed, together with the RAM-disk.

Test Assets

For every test session, we (optionally):

- Record a video of the test, so you can later watch what happened during the test
- Take screenshots of the webpage
- Collect log output from various services (Selenium, our orchestrator and Browser logging)

These assets are uploaded, via SSL, to a protected AWS S3 bucket, located in Ireland.
Each asset is uploaded with “private mode”, meaning nobody can access the asset through public internet.
On our website, we create signed URLs to the assets with a hash that will expire after 30 minutes.

Assets are set to automatically be destroyed from AWS after 31 days. If you delete a test, via our website or through our API, we immediately destroy the test's assets from AWS.

By default TestingBot records a video of your test, together with screenshots and log output. If you don't want this, you can specify an option at the start of your test to complete disable this process. By supplying this option, we will not record anything, either on our website or on our asset provider (AWS).

Firewall

Every test runs on a new, pristine VM. We configure each VM to have its own firewall. This firewall does not allow any inbound connection, other from your test-runner or manual test session. Each test runs on its own, isolated, virtual machine.

If you use our TestingBot Tunnel to access your internal network during testing, we modify the VM's firewall to allow this connection.

TESTINGBOT TUNNEL

Our TestingBot tunnel product allows customers to run tests against websites/apps running on their own computer or network. Typically this tunnel is used to test websites that are not (yet) available via the public internet.

The tunnel consists of these components:

- A component to run on your computer, which is a Java-based application. It consists of a HTTP/s proxy and a SSH client.
- A gateway VM, which is a pristine VM, only accessible to you and your team-members.
- A TestingBot VM for each of your tests, connecting to your computer through the gateway VM.

Tunnel Application

The tunnel Application is written in Java and its source-code is available via https://github.com/testingbot/Testingbot-Tunnel/
You can read more about its features on this page: https://testingbot.com/support/other/tunnel

Gateway VM

Our Tunnel Gateway VMs run a HTTP/s proxy that caches commonly accessed images and scripts, to minimize the traffic going through the tunnel. These cached assets will be destroyed, together with the Gateway VM, as soon as you close the tunnel.
Gateway VMs are configured to only allow connections from your computer. The Gateway will also only allow access from TestingBot VMs running your tests, no other VMs can access the Gateway.

TestingBot VM

The VM will configure the browser used during testing to use the tunnel as a proxy. This way the VM can access the website or app running on your computer or network.

Security

All traffic using the TestingBot Tunnel is sent and received via an SSH tunnel between your computer and the Gateway VM.
The access that the TestingBot Tunnel has on your network is completely within your control. We recommend to run the TestingBot Tunnel in a firewalled DMZ which has access only to those resources required for testing.

You can supply an upstream proxy, specify domains you don't want to proxy and set a custom DNS server.

INFORMATION SECURITY MANAGEMENT

TestingBot has a documented Information Security Management Program which is updated annually. We follow the ITIL v3 framework that guides our IT management practices.

DATACENTER SECURITY

Our datacenter is located in Brussels, Belgium.
All our servers are located in a secure and restricted area, only accessible by a select number of TestingBot employees. The facility provides 24x7 on-site security and monitoring, engineering personnel and password access controls.
The datacenter offers redundant fiber, power and cooling systems.

BUSINESS CONTINUITY MANAGEMENT

We perform nightly backups of our system software to a SAN, replicated to an AWS region. This allows us to recover our data and resume our service in case of a major system failure, in the least amount of time.

We have strict password policies and guidelines for our employees.

INCIDENT RESPONSE

We have an on-call schedule where engineers are responsible to address any issues impacting our service. TestingBot has an extensive monitoring system alerting engineers when our incidents occur. Customers are alerted by system incidents via https://testingbot.statuspage.io/

NETWORK SECURITY

Testingbot runs its service with Ubuntu LTS. We use Ubuntu LTS because it has a good track-record of providing long term support with security patches and upgrades.

Servers are accessible via SSH. Employees access servers through SSH via their own private key.
Servers acting as a hypervisor for the VMs are not connected or accessible over the public internet. These servers are only accessible by TestingBot employees via SSH.

SOFTWARE SECURITY

We regularly scan our websites for SQL injections, XSS bugs and more with a vulnerability scanner. Every six months we schedule a third-party penetration test.

TestingBot makes sure to follow best-practices regarding Software Security, including using only vetted Javascript libraries, using template escaping and more.

CREDENTIALS

Clients of TestingBot can authenticate to the TestingBot service by logging into the TestingBot website with email and password. Users can choose to enforce 2FA (Two-Factor Authentication) during this login process.

Clients can also access the TestingBot API and Grid by providing their private key and secret combination, located in the member area.

SOFTWARE ACCESS

We have policies on who can access what parts of our codebase.
Our service runs different micro-services. Only people/systems that need to have access to a specific micro-service are granted access.

We keep record of all changes to our software. TestingBot employees are required to sign an NDA, to ensure confidentiality of our code and our customer's data.

CONCLUSION

TestingBot provides a secure computing cloud for enterprises looking to run both automated and manual tests on real browsers and platforms. Our service removes the hassle and expense of running and maintaining an internal test grid or lab.

We take security very serious and provide a secure and reliable service, protecting the security and privacy of our customers.

For questions, please contact us at https://testingbot.com/contact/new.

TestingBot.com
Vlinderhof 18
9180 Moerbeke-Waas
Belgium

+32 473 44 24 66