PEACH
FUZZER

Peach Fuzzer Professional Installation Guide

Peach Fuzzer, LLC

Version 0.0.0

Peach Fuzzer, LLC
1122 E Pike St
Suite 1064
Seattle, WA 98122

Table of Contents

1. Preface

This document, the Peach Fuzzer Professional Installation Guide, focuses strictly on installing Peach Fuzzer Professional on all supported operating systems. The same installation instructions can also be found in the Peach Fuzzer Professional User Guide under the "Installation" section. This guide is provided for convenience and does not contain any additional information that is not also present in the Peach Fuzzer Professional User Guide.

2. Installation

The following list contains links to the sections that describe in detail how to download, install and activate Peach Fuzzer.

Recommended Hardware

This section describes the minimum and recommended hardware requirements for common fuzzing scenarios.

Product Download

This section contains the steps required to download the product for your desired platform.

Product Installation

This section lists the software prerequisites, OS specific configuration, and steps for installing the product on each of the three supported platforms:

- Windows
- Linux
- OSX

Product Activation

The list below contains links to the steps for activating the different types of Peach Fuzzer licenses.

- Usage Based (Online Synchronization)
- Usage Based (Offline Synchronization)
- Node Locked
- Enterprise

Optional Configuration

The list below contains links to optional post-install configurations.

- Enabling HTTPS And Authentication

2.1. Hardware Requirements

The following are generic hardware recommendations. Adjust based on your needs.

2.1.1. Local Target

When fuzzing a local target (software running on the same machine as Peach), additional resources are required for the target process.

Target Type	Architecture	Cores	Ram	Disk
Network	64-bit	4	8GB	60GB SSD
File	64-bit	4	16GB	60GB SSD
Other	64-bit	4	8GB	60GB SSD

2.1.2. Remote Target

Remote target fuzzing occurs when the target is not located on the machine running Peach.

	Architecture	Cores	Ram	Disk
Minimum	64-bit	2	4GB	60GB Any
Recommended	64-bit	2	8GB	60GB SSD

It's possible to run Peach on 32-bit systems, but it's not recommended as it places severe limits on memory usage (max 2GB).

2.2. Downloading

The first step of the installation is to download the Peach Fuzzer distribution files from the Peach download site. Once the appropriate files have been downloaded, follow the instructions for your specific operating system found in the next section.

User Account Download

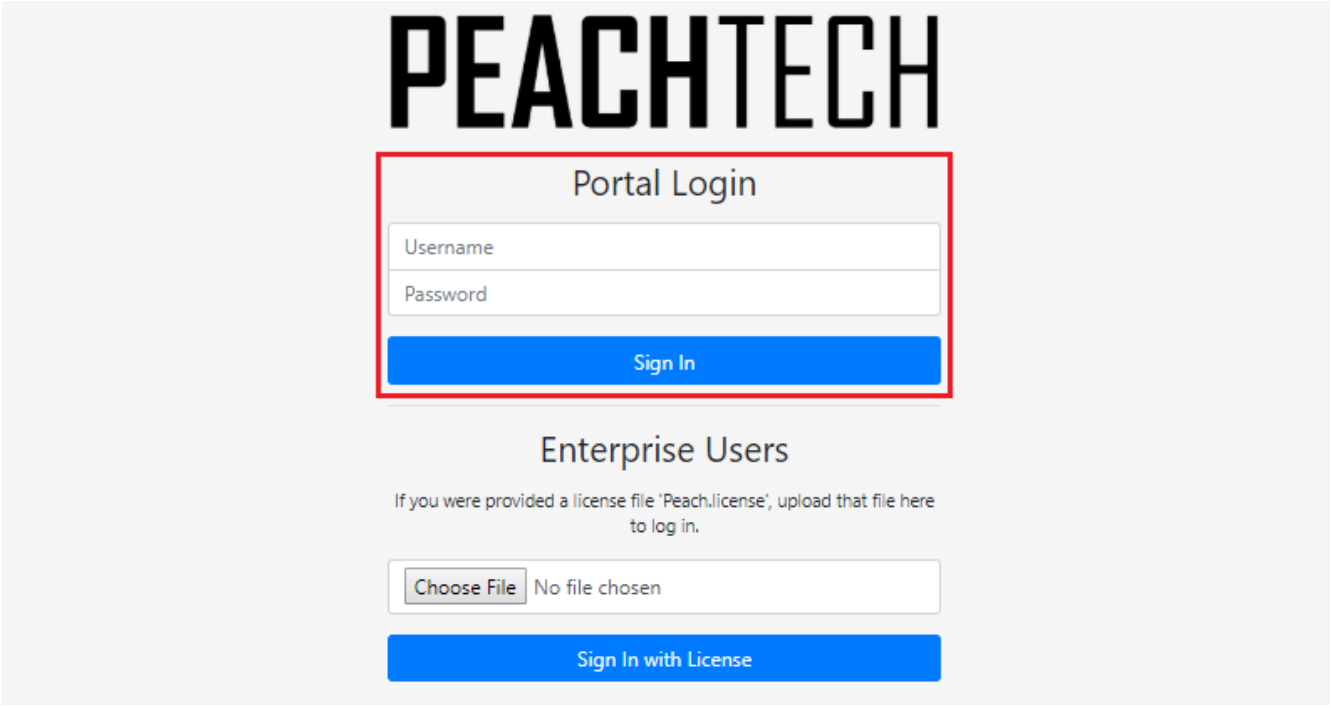
If you were assigned an account with a username/password, follow these instructions to sign in and download Peach Fuzzer.

Enterprise Download

If you were provided an enterprise license file, follow the instructions to sign in and download Peach Fuzzer.

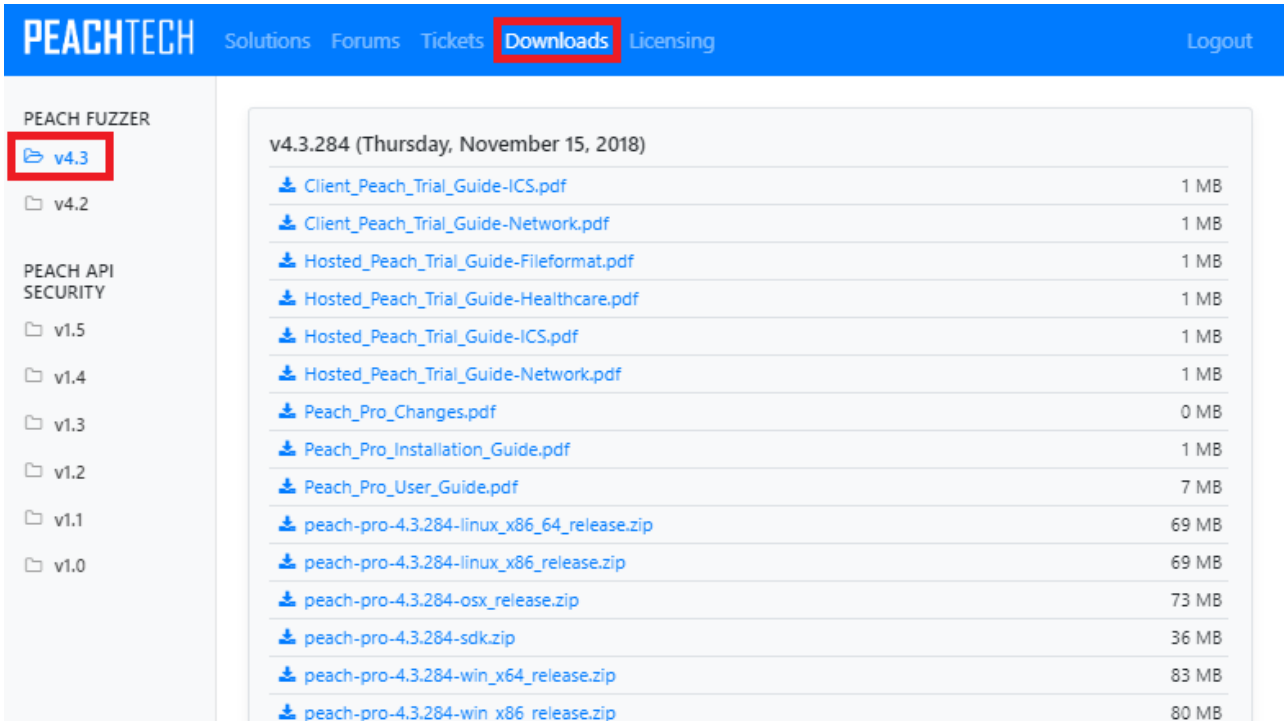
2.2.1. User Account Download

1. When your Peach Fuzzer welcome email arrives, click the link to reset your initial password.
2. Navigate to https://portal.peachfuzzer.com with your preferred web browser.
3. At the login prompt under the Portal Login section, enter the new username/password that was recently reset and click `Sign In`.



Contact support@peach.tech if you need your account password reset.

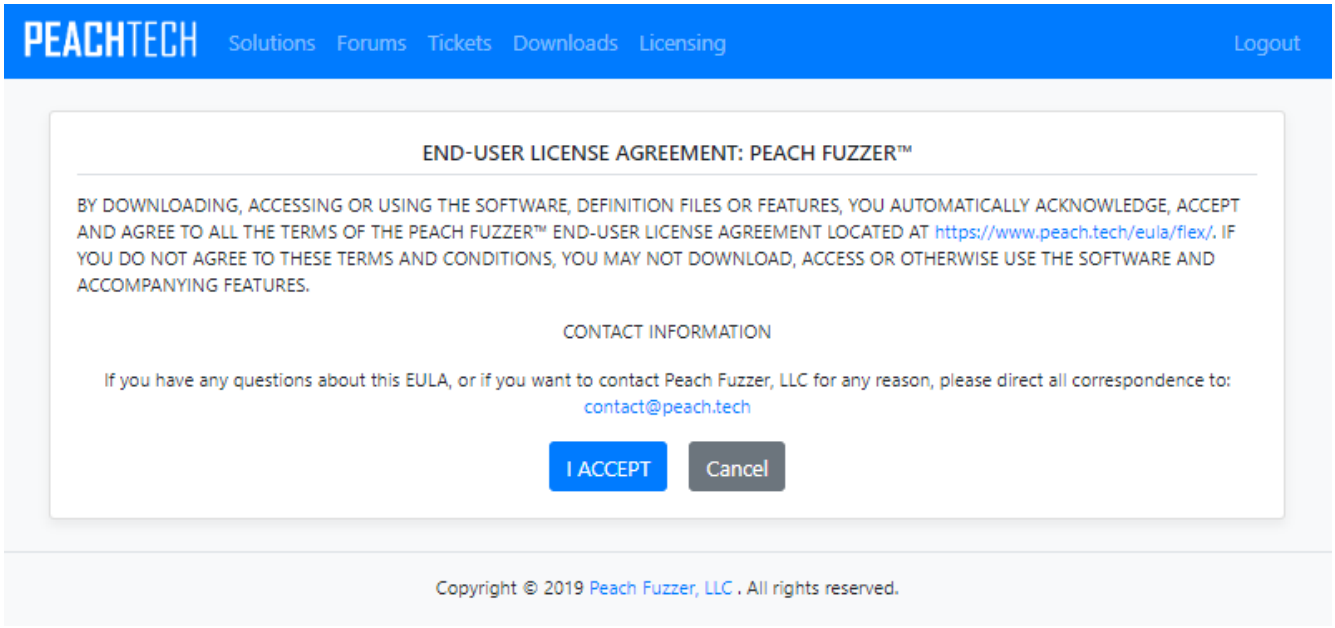
4. On the `Downloads` page, select the Peach Fuzzer release version and operating system to install.
   a. Choose a release version from the items on the left side.
   b. Click the download icon on the right side after deciding which OS and architecture is needed.



5. If your organization has multiple entitlements, you may need to select an Activation ID that corresponds to the license the download should be tied to. This selection page will not be displayed

if there is only one entitlement for your organization. Contact licensing@peach.tech for more information if you are unsure which Activation ID to select.

6. After a few moments, an End User License Agreement acceptance page appears. Click `I ACCEPT` to continue.

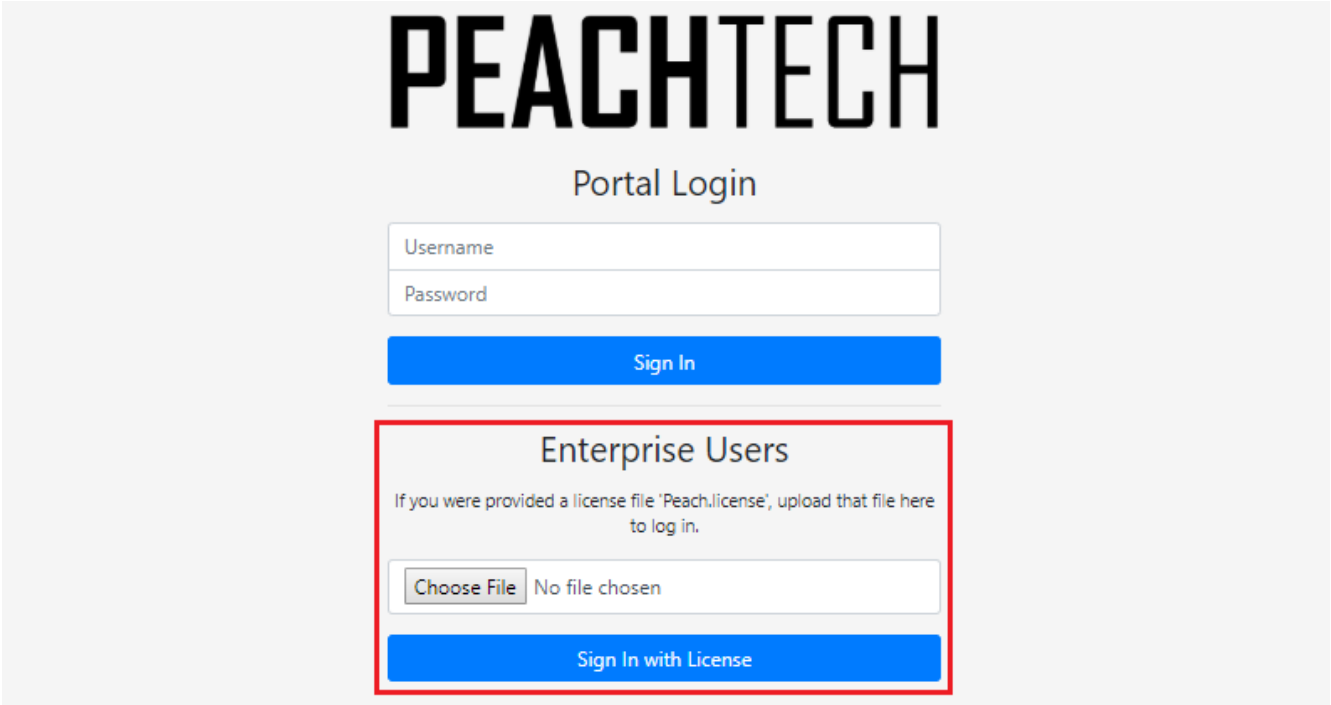


7. The download will begin. Depending on your network connection, this could take a few minutes.

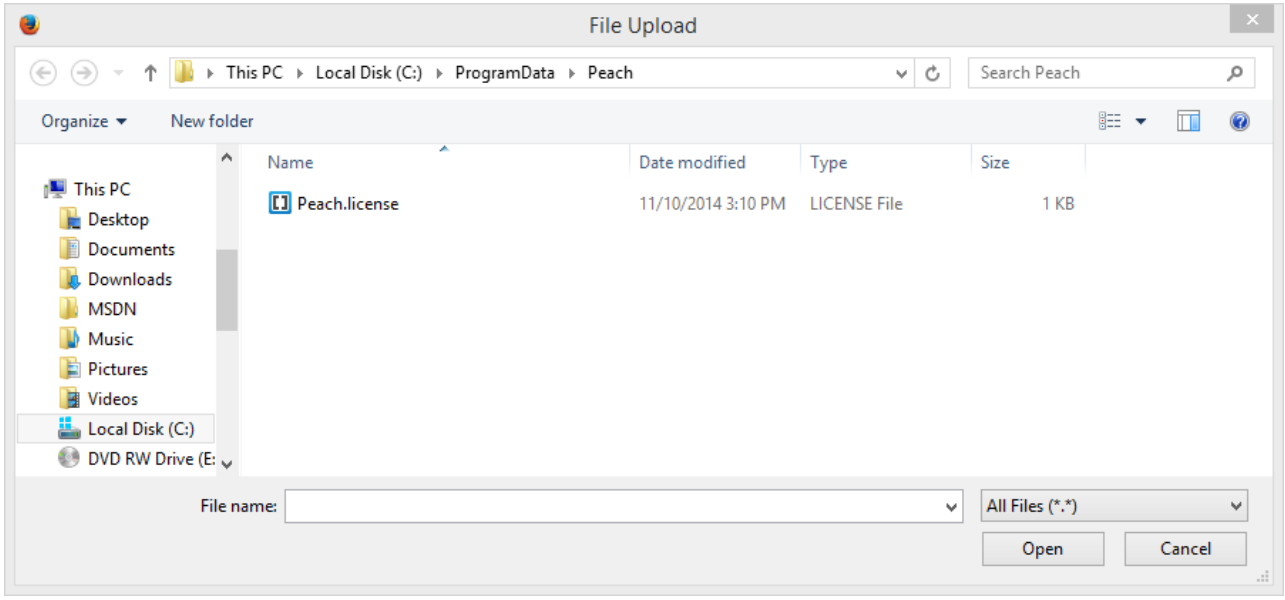
2.2.2. Enterprise Download

You need a copy of your Peach Fuzzer license on your system to perform the download.

1. Using a web browser, navigate to https://portal.peachfuzzer.com



a. Click the `Choose File` button. The upload dialog display allows you to select the `Peach.license` file.



b. Navigate to the location of the license

c. Select the license (`Peach.license`)

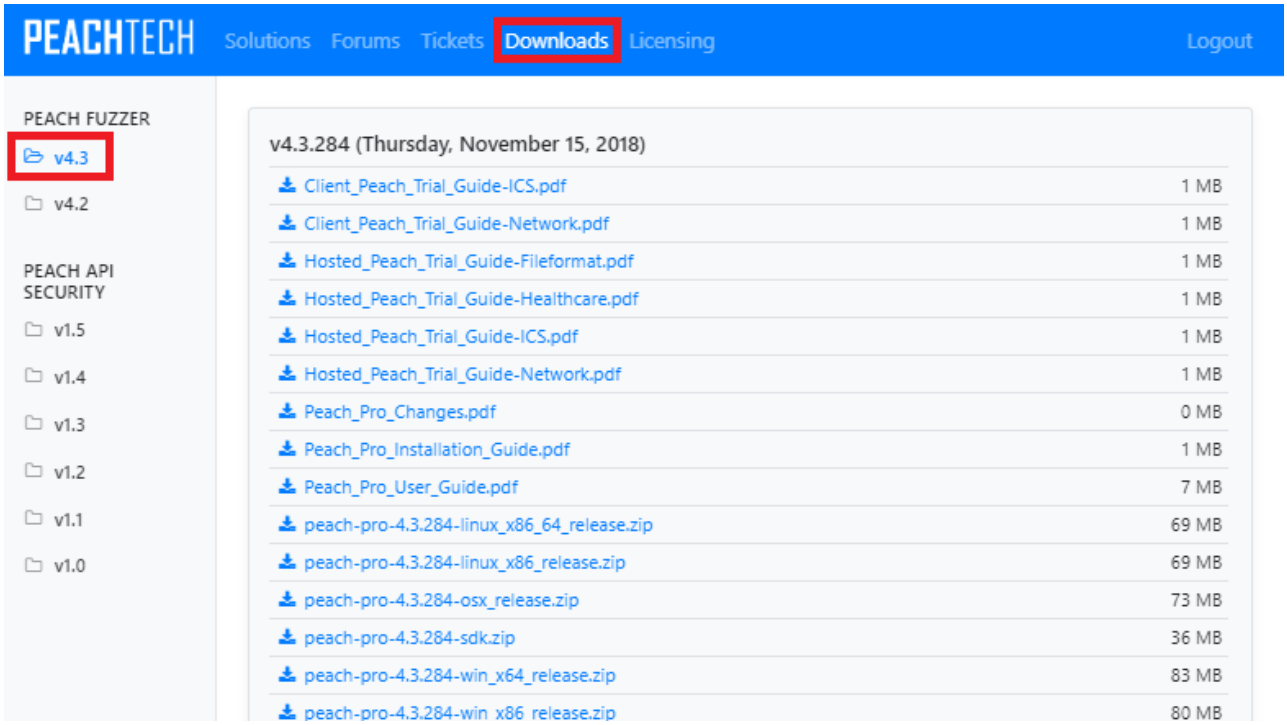
d. Click `Open` to return to the download home page.

2. Upon returning to the Peach download home page, click `Sign In with License`.

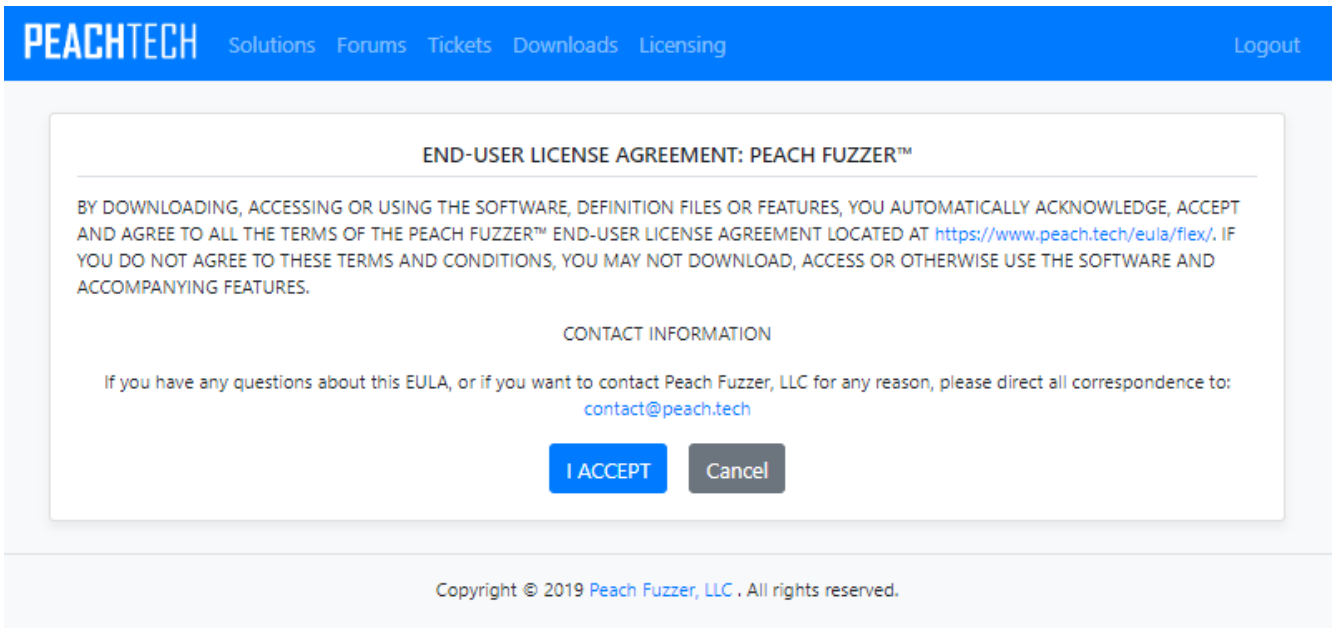
3. On the `Downloads` page, select the Peach Fuzzer release version and operating system to install.

   a. Choose a release version from the items on the left side.

   b. Click the download icon on the right side after deciding which OS and architecture is needed.



4. After a few moments, an End User License Agreement acceptance page appears. Click `I ACCEPT` to continue.



5. The download will begin. Depending on your network connection, this could take a few minutes.

2.3. Windows

Peach is officially supported on the following Windows® Operating Systems:

- Windows 7 SP1 (x86 and x64)
- Windows 8 (x86 and x64)

- Windows 8.1 (x86 and x64)
- Windows 10 (x64)
- Windows Server 2008 SP2 (x86 and x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)
- Windows Server 2012 R2 (x64)

The only required software is the Microsoft .NET Framework v4.5.

1. Download and install the Microsoft .NET Framework v4.5.2 (Installer).
2. Install the Microsoft Debugging Tools for Windows (optional).

This is only required if you want to use a debugger to detect crashes in fuzzed programs.

3. Install Wireshark (optional).

This is only required if you want to collect network captures during fuzzing runs.

4. Unzip the Peach distribution to the appropriate folder. The file is a zip file with the extension `.zip`. Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-win_x64_release.zip`.
5. When fuzzing, many security products (such as anti-virus programs) can interfere or slowdown fuzzing. For network fuzzing, make sure none of the network or host-based network intrusion detection systems (IDS) are running. For file fuzzing, disable anti-virus software; or mark Peach, the target application, and any directories that might have files used in fuzzing, as out of scope for real time monitoring.

2.4. Linux

Peach is supported on three distributions of Linux; Peach may run on other Linux distributions, but are not officially supported. This section provides instructions for installing Peach on the following supported Linux systems, and includes a checklist for installing Peach on other Linux systems:

- Ubuntu/Debian Linux
- Redhat Enterprise Linux (RHEL and CentOS)
- SUSE Enterprise Linux (SLES)

2.4.1. Ubuntu/Debian Linux

When installing Peach on an Ubuntu or Debian Linux system, the operating system is ready for Peach without modification. The installation starts with the Mono .NET runtime, then Peach. If you want to

attach a debugger to a target process, install GDB when the Mono installation completes.

Peach Fuzzer, LLC, recommends using Ubuntu Linux version 16.04 LTS, and Mono .NET runtime version 4.8.1 from the mono project. The mono project has `apt` packages for Ubuntu.

Peach will not run with Mono version 5.0 or newer due to incompatibilities with IronPython. If you have Mono 5.0 installed, you must downgrade to 4.8.1.

Peach will not run with version 4.4 of the Mono runtime as there are known handle leaks which can cause Peach to run out of memory during long fuzzing runs. If you have Mono 4.4 installed, you can either upgrade to 4.6+ or downgrade to 4.2.

Some Linux kernel versions have known issues with the Mono runtime. When using Ubuntu 14.04 LTS, avoid using kernel versions 3.13.0-48 through 3.13.0-54 inclusive. Version 14.04 LTS might require that you update the Linux kernel. If so, perform the following to update the kernel: `sudo apt-get install linux-image-generic`.

The Peach installer checks for compatibility and alerts the user if an incompatibility has been detected.

The following steps will prepare Peach to run properly:

1. Install the latest `mono-complete` package.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF

echo "deb http://download.mono-project.com/repo/debian wheezy/snapshots/4.8.1.0 main"
| sudo tee /etc/apt/sources.list.d/mono-xamarin.list

sudo apt-get update

sudo apt-get install mono-complete
```

2. Install libpcap using the following command:

```
sudo apt-get install libpcap
```

3. Optionally, install the GNU Debugger (GDB) to enable debugging of local processes.

```
sudo apt-get install gdb
```

4. Unzip the Peach distribution to the appropriate folder. The file is a zip file with the extension `.zip`.

Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-linux_x86_64_release.zip`.

2.4.2. Redhat Enterprise Linux (RHEL and CentOS)

Installing Peach on a RHEL CentOS platform requires additional steps. Begin by installing Extra Packages for Enterprise Linux (EPEL), followed by the Mono package and Peach.

Peach Fuzzer, LLC, recommends using Mono .NET runtime version 4.8.1 from the mono project. The mono project has `yum` packages for RHEL and CentOS distributions.

Peach will not run with Mono version 5.0 or newer due to incompatibilities with IronPython. If you have Mono 5.0 installed, you must downgrade to 4.8.1.

Peach will not run with version 4.4 of the Mono runtime as there are known handle leaks which can cause Peach to run out of memory during long fuzzing runs. If you have Mono 4.4 installed, you can either upgrade to 4.6+ or downgrade to 4.2.

The following Mono installation steps are taken from the Mono Project.

The following steps provide the needed details:

1. Install `yum-utils` using the following command:

```
sudo yum install yum-utils
```

2. Install Extra Packages for Enterprise Linux (EPEL) using the following command:

```
sudo yum install epel-release
```

3. Import the GPG signing key for the mono package using the following command. Note the long search key:

```
sudo rpm --import
"http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x3FA7E0328081BFF6A14DA29AA6A19B
38D3D831EF"
```

4. Add and enable the mono project repository for CentOS using the yum configuration manager:

```
sudo yum-config-manager --add-repo http://download.mono-project.com/repo/centos/
```

5. Install the latest version of Mono using the following command:

```
sudo yum install mono-complete-4.8.1.0-0.xamarin.1
```

6. Install libpcap using the following command:

```
sudo yum install libpcap
```

7. Unzip the Peach distribution to the appropriate folder. The file is a zip file with the extension `.zip`. Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-linux_x86_64_release.zip`.

If you receive and error regarding libMonoPosixHelper the `/etc/mono/config` file may need to be edited. To edit locate a line that looks like the following (the path may be different):

```
<dllmap dll="MonoPosixHelper" target="/usr/lib/libMonoPosixHelper.so"
os="!windows" />
```

Once found use the Linux *find* command to locate the shared library:

```
find /usr -name "*libMonoPosixHelper.so"
```

And finally, update the `/etc/mono/config` entry to the correct path.

For more information, see the following resources: * http://www.mono-project.com/docs/getting-started/install/linux#centos-fedora-and-derivatives * https://fedoraproject.org/wiki/EPEL#How_can_I_use_these_extra_packages.3F

2.4.3. SUSE Enterprise Linux (SLES)

To install Peach on a SUSE Enterprise Linux platform, use the 1-click SUSE mono-complete installation file. If you want to attach a debugger to a target process, install GDB when the Mono installation completes.

Peach Fuzzer, LLC, recommends using Mono .NET runtime version 4.8.1 from the mono project. The mono project has packages for SLES distributions.

Peach will not run with Mono version 5.0 or newer due to incompatibilities with IronPython. If you have Mono 5.0 installed, you must downgrade to 4.8.1.

Peach will not run with version 4.4 of the Mono runtime as there are known handle leaks which can cause Peach to run out of memory during long fuzzing runs. If you have Mono 4.4 installed, you can either upgrade to 4.6+ or downgrade to 4.2.

The following steps provide the needed details:

1. Import the GPG signing key for the mono package using the following command. Note the long search key:

```
sudo rpm --import
"http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x3FA7E0328081BFF6A14DA29AA6A19B
38D3D831EF"
```

2. Add and enable the mono project repository using the zypper configuration manager:

```
sudo zypper ar -f http://download.mono-project.com/repo/centos/ mono
```

3. Install the latest supported version of Mono using the following command:

```
sudo zypper in mono-complete=4.8.1.0-0.xamarin.1
```

4. Install libpcap using the following command:

```
sudo zypper in libpcap
```

5. Optionally, install the GNU Debugger (GDB) for debugging local processes.

```
sudo yum install gdb
```

6. Unzip the Peach binary distribution to the appropriate folder. The file is a zip file with the extension `.zip`. Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-linux_x86_64_release.zip`.

If you receive and error regarding libMonoPosixHelper the `/etc/mono/config` file may need to be edited. To edit locate a line that looks like the following (the path may be different):

```
<dllmap dll="MonoPosixHelper" target="/usr/lib/libMonoPosixHelper.so"
os="!windows" />
```

Once found use the Linux *find* command to locate the shared library:

```
find /usr -name "*libMonoPosixHelper.so"
```

And finally, update the `/etc/mono/config` entry to the correct path.

2.4.4. Other Linux Distributions

For other Linux versions, the installation steps are a checklist, not specific commands. The checklist follows:

1. Install the Mono runtime. Version 4.8.1 is recommended.
2. Unzip the Peach distribution to an appropriate folder. The file is a zip file with the extension `.zip`. Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-linux_x86_64_release.zip`.

Peach Fuzzer, LLC, recommends using Mono .NET runtime version 4.8.1 from the mono project.

Peach will not run with Mono version 5.0 or newer due to incompatibilities with IronPython. If you have Mono 5.0 installed, you must downgrade to 4.8.1.

Peach will not run with version 4.4 of the Mono runtime as there are known handle leaks which can cause Peach to run out of memory during long fuzzing runs. If you have Mono 4.4 installed, you can either upgrade to 4.6+ or downgrade to 4.2.

2.5. macOS

To install on macOS, follow the installation steps provided below. Installation will require installing the Mono .NET runtime, then Peach. To enable support for the CrashWrangler monitor, install CrashWrangler and Xcode. Note that installing CrashWrangler is optional; it is only needed when running the target locally.

Peach Fuzzer, LLC, recommends using Mono .NET runtime version 4.8.1 from the mono project.

Peach will not run with Mono version 5.0 or newer due to incompatibilities with our Python runtime. If you have Mono 5.0 installed, you must downgrade to 4.8.1.

Peach will not run with version 4.4 of the Mono runtime as there are known handle leaks which can cause Peach to run out of memory during long fuzzing runs. If you have Mono 4.4 installed, you can either upgrade to 4.6+ or downgrade to 4.2.

1. Install the Mono package.
2. Unzip the Peach distribution to an appropriate folder. The file is a zip file with the extension `.zip`. Use the filename that begins with `peach-pro` and contains the appropriate architecture for your system, such as `peach-pro-0.0.0-osx_release.zip`.
3. Install CrashWrangler.

   CrashWrangler **MUST** be compiled on each macOS machine. Peach includes the CrashWrangler source files in the peach distribution. Here are instructions to install and compile CrashWrangler from the peach zip.

   a. Ensure XCode is installed.

   b. Open `Terminal.app`.

   c. Navigate to the folder where you extracted `peach-pro-0.0.0-osx_release.zip`.

   d. Finish installing CrashWrangler using the following commands.

```
# Navigate to the folder containing CrashWrangler distribution
cd CrashWrangler

# Extract CrashWrangler sources
unzip 52607_crashwrangler.zip

# Navigate to the folder containing the extracted CrashWrangler sources
cd crashwrangler

# Compile CrashWrangler
$ make

# Ensure installation directory exists
sudo mkdir -p /usr/local/bin

# Install CrashWrangler
sudo cp exc_handler /usr/local/bin

# Navigate to the folder containing peach
cd ../../

# Verify CrashWrangler can run
exc_handler
```

2.6. License Activation

The list below contains links to the steps for activating the different types of Peach Fuzzer licenses.

Usage Based (Online Synchronization)

The most common method for activating a usage based license. A Cloud License Server will be automatically provisioned and managed for you. However, Peach Fuzzer will require a persistent connection to the Internet.

Usage Based (Offline Synchronization)

Users who wish to use Peach Fuzzer in an offline without having access to the Internet can deploy a Local License Server onsite to provide offline activation and synchronization of licensing information.

Node Locked

The license is tied to the physical machine running Peach Fuzzer. No license server is required and internet connectivity is only needed for activation.

Enterprise

No activation is required for enterprise customers.

2.6.1. Usage Based (Online Synchronization)

A Cloud License Server provides functionality for serving and monitoring a counted pool of licenses for Peach Fuzzer. A persistent connection to the Internet is required so that usage data can be uploaded to the Cloud License Server while a Peach Fuzzer job is running. Peach Fuzzer will automatically activate the first time it is run.

If a proxy server is required to connect to the Internet, it must be configured as described below.

Windows Proxy configuration

On Windows, the system proxy setting is the correct way to configure the proxy peach will use to connect to the licensing server.

Windows 10

The system proxy settings are configured at Settings > Network & Internet > Proxy. From there you will be able to enter the IP and Port of the proxy server.

Windows 8

The system proxy settings are configured at PC Settings > Network Proxy. From there you will be able to enter the IP and Port of the proxy server.

Windows 7

The system proxy settings are configured through the Internet Settings dialog. Open the Internet Options window located at Control Panel > Network and Internet > Internet Options.

1. Click the "Connections" tab at the top of the Internet Options window.
2. Click the "LAN Settings" button at the bottom of the window.
3. Click the "Advanced" button under Proxy Server will allow you to change advanced settings and enable a manual proxy server.

Linux Proxy Configuration

The proxy configuration on Linux is controlled via two environment variables `http_proxy` and `https_proxy`. Ensure both variables are set prior to starting Peach Fuzzer.

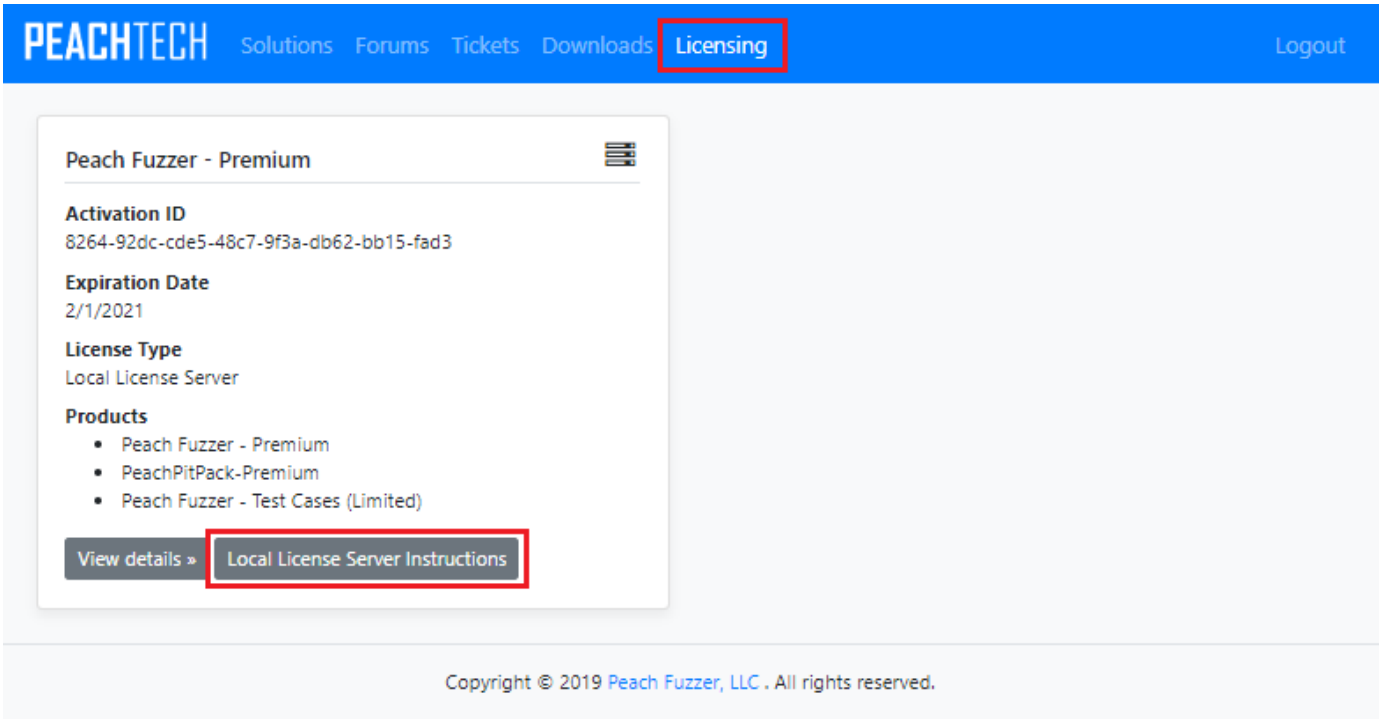
```
export http_proxy=http://xxxxx
export https_proxy=https://xxxxx
```

2.6.2. Usage Based (Offline Synchronization)

The Local License Server provides functionality for serving and monitoring a counted pool of licenses for Peach Fuzzer. Users who wish to use Peach Fuzzer without having access to the Internet can deploy

a Local License Server onsite to provide offline activation and synchronization of licensing information.

The instructions for installing and activating a Local License Server can be found on the Peach Portal by navigating to the "Licensing" tab and clicking the "Local License Server" button for the desired license.



2.6.3. Node Locked

No license server is required, as node locked licenses are tied to an individual machine. Peach Fuzzer will automatically activate the first time it is run.

If your license has changed and you want Peach Fuzzer refresh its license, run the following command:

```
peach --activate
```

If you wish to move your license to a new machine, you must first deactivate the existing instance by running the following command:

```
peach --deactivate
```

Peach Fuzzer requires an internet connection in order to perform activation and deactivation. Once activated, no further internet connectivity is required.

2.6.4. Enterprise

No activation is required for enterprise customers. The enterprise license is automatically embedded in the Peach Fuzzer download.

2.7. Enabling HTTPS And Authentication

Peach Fuzzer Professional uses a web interface for configuration and control of the fuzzing engine. By default, the web interface is accessible with no encryption (SSL/TLS) and no authentication. If the use of HTTPS or authentication is required, a reverse proxy (apache/nginx/traefik) can be used to provide both SSL/TLS and authentication.

2.7.1. Reverse Proxy with NGINX

The following steps will configure NGINX as a reverse proxy for Peach adding TLS and authentication:

1. Install nginx using your Linux package manager
2. Create required key. For self signed keys this online self-signed certificate generator can be used.
3. Install the included NGINX configuration file to /etc/nginx/sites-available/peach
4. Install certificate and key and update configuration file if needed
5. Create .htpasswd with username/passwords replacing `USERNAME` with your username

```
sudo sh -c "echo -n 'USERNAME:' >> /etc/nginx/.htpasswd"
sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"
```

6. Add a firewall rule to block external access to Peach's port 8888. Make sure this rule is enabled on bootup.
7. Link /etc/nginx/sites-available/peach to /etc/nginx/sites-enabled/peach
8. Restart NGINX and verify configuration is working

NGINX Configuration File

```
# HTTPS server
#
server {
        listen 443;
        server_name localhost;

        root html;
        index index.html index.htm;

        ssl on;
        ssl_certificate /etc/ssl/certs/ssl.crt;
        ssl_certificate_key /etc/ssl/private/ssl.key;

        ssl_session_timeout 5m;

        ssl_protocols TLSv1.1 TLSv1.2;
        ssl_ciphers "HIGH:!aNULL:!MD5 or HIGH:!aNULL:!MD5:!3DES";
        ssl_prefer_server_ciphers on;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #try_files $uri $uri/ =404;
                # Uncomment to enable naxsi on this location
                # include /etc/nginx/naxsi.rules


                auth_basic "Restricted";
                auth_basic_user_file /etc/nginx/.htpasswd;
                proxy_pass http://127.0.0.1:8888/;
        }
}
```