PEACH
FUZZER

# PEACH PIT LIBRARY

Q2 2014

Déjà vu Security, LLC
1122 E Pike St
Suite 1071
Seattle, WA 98112

# Contents

# 1 Preface

This book is the official documentation for the Peach Pit Library. It has been written by the Peach Fuzzer team and represents a concerted effort to fully document all of the Peach Pit Library features and configurations. Peach Fuzzer has been in active development through three major revisions since 2004. Documenting Peach is an on-going effort. The majority of effort has been placed in documenting the most common uses of Peach Pit Library.

## 1.1 What is Peach Pit Library?

Peach Pit Library is a collection of fuzzing definitions (pits) for the commercial versions of Peach Fuzzer. The collections of pits is updated quarterly with new definitions and update to existing definitions. The updates to the library coincide with updates to Peach Fuzzer and should be considered a matching pair.

## 1.2 Further Information

Further information about Peach can be found on our website and also the user forums.

## 1.3 Bug Reporting Guidelines

Support for Peach Pit Library is available in two ways:

- The Peach Forums site
- Using our ticketing system to open a support ticket

### 1.3.1 Peach Forums

There are two sets of forums for Peach, the community forums and the professional forums. Both forums are hosted at https://forums.peachfuzzer.com. Peach Pit Library users should access the private forums to receive much better response time. To access the professional forums, first create an account on the forums site then send an email to peach@dejavusecurity.com with your license email, and forum username. Your account will be granted access to the professional forums within 24 hours during the business week. Forums are monitored by the Deja vu Security team, but there is no guarantee of response time.

When posting please including the following information:

- Operating system(s) in use by Peach and any agents
- Exact version of Peach being used. This is available from the console output and in the *status.txt* log file.
- Version of Peach Pit Library in use. This version is "Q2 2014".
- Detailed description of the issue and expected behavior
- Console output using the *--trace* argument
- Configuration files and, if modified, the pit file.

### 1.3.2 Support Tickets

To open a support ticket send an email to *support@dejavusecurity.com*. You will receive an initial response within 24 business hours of opening the ticket. Peach support is available Monday through Friday. Peach support is not currently available on the weekends or holidays. When opening a ticket, please provide the following information in your email:

- Operating system(s) in use by Peach and any agents

- Exact version of Peach being used. This is available from the console output and in the *status.txt* log file.
- Detailed description of the issue and expected behavior
- Console output using the *--trace* argument
- (if possible) the full Pit file + configuration files

# 2 Peach Pit Library

The following sections provide usage guides for each pit included in the library.

The fuzzing definitions in this collection are complete definitions, however they will need to be configured for you target. Part of this configuration will be modifying the agent and monitor configuration to suite your target environment. The main Peach documentation includes sections on agents and monitors. A sample configuration is provided.

Agents and monitors are how Peach is able to detect a fault, collect interesting information, and also perform automation with the target environment. This can include attaching debuggers, restarting virtual machines, or running commands via SSH.

# 3 Installation

The Peach Pit Library is distributed as an archive containing all of the files required. The archive should be expanded into a folder called `pits` located in the same folder that Peach Professional has been installed to.

1. Unarchive the library to a folder named `pits`
2. Place the `pits` folder into the Peach Professional folder (contains `Peach.exe`)

# 4 Quickstart

After installing the Peach Pit Library, launch Peach.exe and browse to the indicated port number using a recent version of Chrome, Firefox, or IE. Through the Peach Web UI you can select a fuzzing definition (Pit), configure it and start fuzzing. The Pits can also be edited manually and run from the command line.

# 5 Address Resolution Protocol (ARP)

- Peach Pit: ARP
- Direction: Broadcast, Listen
- Supported Platforms: Linux

The Address Resolution Protocol translates between hardware and protocol addresses. In Ethernet and Wireless networks, it usually translates between IPv4 and MAC addresses.

## 5.1 Specifications

| Specification | Title |
|---|---|
| RFC826 | An Ethernet Address Resolution ProtocolS |

## 5.2 Use Cases

| Messages | Specification |
|---|---|
| MSG 1 | RFC826 |

| Supported Features | Specification |
|---|---|
| Arp Request Packet Generation for IPv4 over Ether | RFC826 |
| Arp Reply Packet Generation in response to a request for IPv4 over Ether | RFC826 |

## 5.3 Configuration

### 5.3.1 Target Configuration

This pit sends ARP packets; no extra applications are required.

### 5.3.2 Required Pit Configuration Changes

**TargetIPv4**
IPv4 address of the target host machine.

**SourceIPv4**
IPv4 address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface (used for monitoring).

#### 5.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 5.3.4 Configure Additional Monitors

Add monitors to agent as needed

### 5.4 Running

#### 5.4.1 Single test debug run

**Fuzzing ARP Request**

```
peach -1 --debug ARP.xml
```

**Fuzzing ARP Reply**

```
peach -1 --debug ARP_Reply.xml
```

#### 5.4.2 Full test run

**Fuzzing ARP Request**

```
peach ARP.xml
```

**Fuzzing ARP Reply**

```
peach ARP_Reply.xml
```

### 5.5 Examples

**Example 5.1** Sample ARP Request Configuration File

Example configuration to broadcast arp packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
  <All>
    <Hwaddr             key="SourceMAC"
                        value="5254005335d3"
                        name="Source MAC Address"
                        description="Hardware address of the network interface on ↩
                            machine running Peach Fuzzer. To find the hardware ↩
                            address on Windows, run 'ipconfig /all' and look for the ↩
                             'Physical Address' field. For Linux run 'ifconfig' and ↩
                            look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                             look for the 'ether' field."/>
```

```
    <Ipv4                key="SourceIPv4"
                                value="127.0.0.1"
                                name="Source IP Address"
                                description="The IPv4 address of the machine running Peach ↩
                                    Fuzzer. The IPv4 address can be found on Windows by ↩
                                    running 'ipconfig' and looking for the 'IPv4 Address' ↩
                                    field. For Linux run 'ifconfig' and look for 'inet addr' ↩
                                     field. For OS X run 'ifconfig' and look for the 'inet' ↩
                                    field."/>
    <Hwaddr              key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
                                description="Hardware address of the network interface on ↩
                                    target machine or device. To find the hardware address ↩
                                    on Windows, run 'ipconfig /all' and look for the ' ↩
                                    Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."  />

    <Ipv4                key="TargetIPv4"
                                value="127.0.0.2"
                                name="Target IP Address"
                                description="The IPv4 address of the target machine or ↩
                                    device. The IPv4 address can be found on Windows by ↩
                                    running 'ipconfig' and looking for the 'IPv4 Address' ↩
                                    field. For Linux run 'ifconfig' and look for 'inet addr' ↩
                                     field. For OS X run 'ifconfig' and look for the 'inet' ↩
                                    field." />

        <String          key="LoggerPath"
                                value="logs/arp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log ↩
                                    produced when fuzzing." />
    <Strategy    key="Strategy"
                                value="Random"
                                name="Mutation Strategy"
                                description="The mutation strategy to use when fuzzing." />

    <String              key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
  </All>

  <Linux>
    <Iface               key="Interface"
                                value="eth0"
                                name="Network Interface"
                                description="The network interface to transmit packets over ↩
                                    . For Windows, the network interfaces can be shown by ↩
                                    running 'ipconfig'. On Linux and OS X, the network ↩
                                    interfaces can be shown by running the command 'ifconfig ↩
                                    '."/>
  </Linux>

  <OSX>
    <!-- This can't run on OS X because it uses the RawEther publisher. -->
  </OSX>

  <Windows>
    <!-- This can't run on windows because it uses the RawEther publisher. -->
```

```
  </Windows>
</PitDefines>
```

# 6 Audio Video Interleaved (AVI)

- Peach Pit: avi_divx
- Supported Platforms: Windows, Linux, OS X

Audio Video Interleaved, AVI, is a multimedia container format introduced by Microsoft part of its Video for Windows technology.

AVI files can contain both audio and video data in a file container that allows synchronous audio-with-video playback.

## 6.1 Specifications

| Specification | Title |
|---|---|
| http://msdn.microsoft.com/en-us/library/windows/-desktop/dd318189(v=vs.85).aspx | AVI RIFF File Reference |

## 6.2 Configuration

### 6.2.1 Target Configuration

A video player that supports the avi format is required. The program VLC can be used for this.

### 6.2.2 Required Pit Configuration Changes

**Seed**
Name of a valid avi file located in the SamplePath directory. An empty string indicates use all files in the directory.

**SamplePath**
Directory path to the directory which the Avi sample files are stored.

**Target**
The program that will open the fuzzed Avi files

### 6.2.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

**Agent**
Agent to run depending on the target OS. This value shouldn't be changed.

### 6.2.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 6.3 Running

### 6.3.1 Single test debug run

```
peach -1 --debug avi_divx.xml
```

### 6.3.2 Full test run

```
peach avi_divx.xml
```

## 6.4 Examples

**Example 6.1** Sample avi_divx Configuration File

Example configuration using VLC player.
First, install VLC player; for this example we assume you are running on Ubuntu or Debian. For other platforms follow instructions on the VLC website:

```
sudo apt-get install vlc
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="Seed"
                                value="avi_divx_sample*.avi"
                                name="Seed File"
                                description="The name of the sample file to use when ↩
                                    fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing." ↩
                                      />

                <String key="LoggerPath"
                                value="./logs/avi_divx/"
                                name="Logger Path"
                                description="The directory where Peach will save the log ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="./_Common/Samples/Video"
                                name="Sample Path"
                                description="The directory containing the samples to use ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see VLC player repeatedly open and close.

# 7 BMP Image Format

- Peach Pit: BMP
- Supported Platforms: Windows, Linux, OS X

Bitmap (BMP) is a Microsoft defined file format containing raster graphic images. Bitmap files contain fixed sized headers and a variable length pixel array.

The BMP header varies by a the version number; each subsequent version appends fields to end of the previous version. For example:

- The v2 header is the same header as the v1 header except that there are additional data fields appended to the v1 header
- The v3 is header is the same header as the v2 header except that there are additional data fields appended to the v2 header

The file format is closely tied to the DIB internal data structure in the Windows API.

## 7.1 Specifications

| Specification | Title |
|---|---|
| http://msdn.microsoft.com/en-us/library/-dd183386%28VS.85%29.aspx | Bitmap Header Types |

## 7.2 Use Cases

| Supported Headers | Specification |
|---|---|
| BITMAPCOREHEADER | Bitmap Header Types |
| BITMAPINFOHEADER | Bitmap Header Types |
| BITMAPV4HEADER | Bitmap Header Types |
| BITMAPV5HEADER | Bitmap Header Types |

## 7.3 Configuration

### 7.3.1 Target Configuration

The BMP file format can target any number of image viewing programs (such as "feh" on Linux and "mspaint.exe" on Windows) by setting the target program that you are fuzzing (such as feh or mspaint.exe) in the Bmp.xml.config pit file.

Normally you would set different target programs for different operating systems.

### 7.3.2 Required Pit Configuration Changes

**Seed**
: Name of a valid BMP file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together with an asterisk wildcard.

**Target**
: The program that opens the fuzzed BMP files

#### 7.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the Bmp sample files are stored. Relative to Path.

#### 7.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 7.4 Running

#### 7.4.1 Single test debug run

```
peach -1 --debug Bmp.xml
```

#### 7.4.2 Full test run

```
peach Bmp.xml
```

### 7.5 Examples

**Example 7.1** Sample Bmp Configuration File

This example configuration uses feh on Linux. The configuration file also contains settings for mspaint on Windows and preview on OSX:
First, install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.bmp"
                                name="Fuzzed Output File"
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="*.bmp"
```

```
                                name="Seed File"
                                description="The name of the sample file to use when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing." />

                <String key="LoggerPath"
                                value="##PitLibraryPath##/logs/bmp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this, feh will repeatedly open and close.

# 8 Cisco Discovery Protocol

- Peach Pit: CDP
- Direction: Announce
- Supported Platforms: Linux

The Cisco Discovery Protocol (CDP) is a proprietary Data Link Layer and Network Layer protocol developed by Cisco Systems. Cisco's multicast announce based protocol is used for neighbor device discovery. It is supported on numerous Cisco devices as well as many devices designed to inter-operate with Cisco hardware (primarily routers and switches).

This protocol provides similar functionality to vendor-neutral IEEE 802.1AB Link Layer Discovery Protocol designed to replace.

Cisco documentation (once found at http://www.cisco.com/univercd/cc/td/doc/product/lan/trsrb/frames.htm) was used to create this pit. This documentation has since been removed by Cisco; no official documentation is available for reference.

## 8.1 Specifications

| Specification | Title |
|---|---|
| http://www.cisco.com/univercd/cc/td/doc/product/lan/-trsrb/frames.htm | CDP Packet Format |

## 8.2 Use Cases

| Messages | Specification |
|---|---|
| Announce | http://www.cisco.com/univercd/cc/td/doc/product/lan/-trsrb/frames.htm |

## 8.3 Configuration

### 8.3.1 Target Configuration

This pit broadcasts CDP packets; no extra applications are required.

### 8.3.2 Required Pit Configuration Changes

**SourceIPv4**
IPv4 address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface.

**Hostname**
Hostname of the switch.

#### 8.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 8.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 8.4 Running

#### 8.4.1 Single test debug run

```
peach -1 -debug CDP.xml
```

#### 8.4.2 Full test run

```
peach CDP.xml
```

### 8.5 Examples

**Example 8.1** Sample CDP Configuration File

Example configuration to broadcast CDP packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="Hostname"
                                value="foobar"
                                name="Hostname"
                                description="Hostname to provide via CDP protocol."/>
                <String key="Domain"
                                value="foobar"
                                name="Domain"
                                description="Domainname to provide via CDP protocol."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
```

```
                        name="Source IPv4 Address"
                        description="The IPv4 address of the machine running Peach Fuzzer ↩
                            . The IPv4 address can be found on Windows by running ' ↩
                            ipconfig' and looking for the 'IPv4 Address' field. For Linux ↩
                            run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet' field."/>

                <Hwaddr key="TargetMAC"
                            value="000000000000"
                            name="Target MAC Address"
                            description="Hardware address of the network interface on ↩
                                target machine or device. To find the hardware address ↩
                                on Windows, run 'ipconfig /all' and look for the ' ↩
                                Physical Address' field. For Linux run 'ifconfig' and ↩
                                look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                 look for the 'ether' field." />

                <Hwaddr key="SwitchMAC"
                            value="000000000000"
                            name="Target MAC Address"
                            description="Hardware address of the network interface on ↩
                                target machine or device. To find the hardware address ↩
                                on Windows, run 'ipconfig /all' and look for the ' ↩
                                Physical Address' field. For Linux run 'ifconfig' and ↩
                                look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                 look for the 'ether' field." />

                <String key="LoggerPath"
                            value="logs/cdp/"
                            name="Logger Path"
                            description="The directory where Peach will save the log ↩
                                produced when fuzzing." />

                <Strategy key="Strategy"
                              value="Random"
                              name="Mutation Strategy"
                              description="The mutation strategy to use when fuzzing." ↩
                                  />

                <String key="PitLibraryPath"
                            value="."
                            name="Pit Library Path"
                            description="The path to the root of the pit library."/>
        </All>

        <Linux>
                <Iface key="Interface"
                           value="eth0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be ↩
                               shown by running the command 'ifconfig'."/>
        </Linux>

        <OSX>
                <Iface key="Interface"
                           value="en0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be ↩
```

```
                                shown by running the command 'ifconfig'."/>
        </OSX>

        <Windows>
                <Iface key="Interface"
                            value="Local Area Connection"
                            name="Network Interface"
                            description="The network interface to transmit packets over. For ↩
                                 Windows, the network interfaces can be shown by running ' ↩
                                ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                                shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 9 Dynamic Host Configuration Protocol version 4 (DHCPv4)

- Peach Pit: DHCPv4
- Direction: Client
- Supported Platforms: Linux

Dynamic Host Configuration Protocol Version 4 (DHCPv4) is the network configuration protocol over IPv4.

DHCP provides a way for devices on a network to request an available IP address from a central server. Requesting IP addresses with DHCP reduces the changes of address collisions and allows hosts to become addressable without needing the network configuration's previous details.

Requests are broadcasted over the layer-2 network to an entire network space and relayed on until received by a DHCP Server. The server responds back with an available IP address and a handshake is completed between the two systems.

## 9.1 Specifications

| Specification | Title |
|---|---|
| RFC1531 | Dynamic Host Configuration Protocol |
| RFC1541 | Dynamic Host Configuration Protocol |
| RFC2131 | Dynamic Host Configuration Protocol |
| RFC3396 | Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4) |

## 9.2 Use Cases

| Messages | Specification |
|---|---|
| DHCP Message | RFC2131 (Section 2) |

| Supported Features | Specification |
|---|---|
| DHCPREQUEST | RFC2131 (Section 4.3.2) |
| DHCPACK | RFC2131 (Section 4.4) |

## 9.3 Configuration

### 9.3.1 Target Configuration

A target machine with DHCPv4 enabled. The software tool bind can be targeted in a Linux environment. A firewall must not block the DHCPv4 messages.

### 9.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**SourcePort**
UDPv4 port number of the local machine.

**SourceMAC**
MAC Address of the local machine.

**TargetMAC**
MAC Address of the target host machine.

**Interface**
Name of local interface.

#### 9.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 9.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 9.4 Running

#### 9.4.1 Single test debug run

```
peach -1 --debug DHCPv4.xml
```

#### 9.4.2 Full test run

```
peach DHCPv4.xml
```

### 9.5 Example

**Example 9.1** Sample DHCPv4 Configuration File

Example configuration sending DHCPv4 packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Dynamic Host Configuration Protocol (DHCP-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
```

```
                        description="Hardware address of the network interface on machine running Peach Fuzzer. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."/>

        <Ipv4 key="SourceIPv4"
                  value="0.0.0.0"
                  name="Source IPv4 Address"
                  description="This is an advanced option and should be left as default."/>

        <Range key="SourcePort"
                   value="1055"
                   min="0" max="65535"
                   name="Source Port"
                   description="The source port the network packet originates from."/>

        <Hwaddr key="TargetMAC"
                        value="000000000000"
                        name="Target MAC Address"
                        description="Hardware address of the network interface on target machine or device. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field." />

        <Ipv4 key="TargetIPv4"
                  value="255.255.255.255"
                  name="Target IPv4 Address"
                  description="This is an advanced option and should be left as default." />

        <Range key="TargetPort"
                   value="67"
                   min="0" max="65535"
                   name="Target Port"
                   description="The destination port the network packet is sent to."/>

        <String key="LoggerPath"
                        value="logs/dhcpv4/"
                        name="Logger Path"
                        description="The directory where Peach will save the log produced when fuzzing." />

        <Strategy key="Strategy"
                          value="Random"
                          name="Mutation Strategy"
                          description="The mutation strategy to use when fuzzing." />

        <String key="PitLibraryPath"
                        value="."
                        name="Pit Library Path"
                        description="The path to the root of the pit library."/>
    </All>

    <Linux>
```

```
                <Iface key="Interface"
                           value="eth0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </Linux>

        <OSX>
                <Iface key="Interface"
                           value="en0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </OSX>

        <Windows>
                <Iface key="Interface"
                           value="Local Area Connection"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 10 Dynamic Host Configuration Protocol version 6 (DHCPv6)

- Peach Pit: DHCPv6
- Direction: Client, Server
- Supported Platforms: Windows, Linux, OS X

Dynamic Host Configuration Protocol Version 6 (DHCPv6) is a variation of the DHCPv4 protocol for the IPv6 address space.

- Like DHCPv4, DHCPv6 allocates addresses from a central server and requests are broadcast across the link layer.
- Unlike IPv4, hosts on an IPv6 network can automatically assign address without DHCP or a central server with stateless address auto-configuration.

Hosts may still use DHCPv6 even if they are not requesting an address to configure other network parameters (like DNS settings).

## 10.1 Specifications

| Specification | Title |
|---|---|
| RFC3315 | Dynamic Host Configuration Protocol for IPv6 (DHCPv6) |
| RFC3633 | IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 |
| RFC3736 | Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6 |
| RFC6221 | Lightweight DHCPv6 Relay Agent |
| RFC6422 | Relay-Supplied DHCP Options |
| RFC6644 | Rebind Capability in DHCPv6 Reconfigure Messages |

## 10.2 Use Cases

| Messages | Specification |
|---|---|
| DHCPv6 Message | RFC3315 (Section 6) |
| Relay Agent Message | RFC3315 (Section 7) |

| Supported Features | Specification |
|---|---|
| SOLICIT | RFC3315 (Section 5.3) |
| ADVERTISE | RFC3315 (Section 5.3) |
| REQUEST | RFC3315 (Section 5.3) |
| CONFIRM | RFC3315 (Section 5.3) |
| RENEW | RFC3315 (Section 5.3) |
| REBIND | RFC3315 (Section 5.3) |
| REPLY | RFC3315 (Section 5.3) |
| RELEASE | RFC3315 (Section 5.3) |
| DECLINE | RFC3315 (Section 5.3) |
| RECONFIGURE | RFC3315 (Section 5.3) |
| INFORMATION-REQUEST | RFC3315 (Section 5.3) |
| RELAY-FORW | RFC3315 (Section 5.3) |
| RELAY-REPL | RFC3315 (Section 5.3) |

## 10.3 Configuration

### 10.3.1 Target Configuration

A target machine with DHCPv6 enabled. The software tool bind6 can be targeted in a Linux environment. A firewall must not be blocking DHCPv6 messages.

### 10.3.2 Required Pit Configuration Changes

The following options must be updated with the correct information.

**TargetIPv6**
IP address of the DHCP server (target)

**TargetMAC**
MAC address of the DHCP server (target)

**SourceIPv6**
IP address of the interface on the local machine (source). Scope/link address is acceptable.

**SourceMAC**
MAC address of the local machine (source)

### 10.3.3 Optional Pit Configuration Changes

The following options have defaults that are acceptable.

**TargetIPv6Lease**
Target IPv6 Lease (fuzzed in DataModel)

**MaxTries**
Maximum number of packets to listen for before going to next iteration

**SourcePort**
UDPv6 port number of the local machine.

**TargetPort**
UDPv6 port number of the target host machine

**Strategy**
Fuzzing strategy Peach will use for testing

**LoggerPath**
Path to folder where logs will be stored

**PitLibraryPath**
Path to the relative base directory where all pits are located.

### 10.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 10.4 Running

### 10.4.1 Single test debug run

**Fuzzing DHCPv6 Server**

```
peach -1 -debug DHCPv6_Server.xml
```

**Fuzzing DHCPv6 Client**

```
peach -1 -debug DHCPv6_Client.xml
```

### 10.4.2 Single test debug run

**Fuzzing DHCPv6 Server**

```
peach DHCPv6_Server.xml
```

**Fuzzing DHCPv6 Client**

```
peach DHCPv6_Client.xml
```

## 10.5 Examples

**Example 10.1** Sample DHCPv6 Client Configuration File

Example configuration sending DHCPv6 packets.
For this example we assume you are running on Ubuntu or Debian Linux.
**Install DHCPv6 Client**

```
sudo apt-get install wide-dhcpv6-client
```

**Running the client**

```
dhcpv6c -f <interface>
```

```
NOTE: The MAC addresses must be updated based on the environment.
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="TargetIPv6Lease"
                                value="fe80::908:7a38:6156:acce"
                                name="Target IPv6 Lease"
                                description="IPv6 address to assign."/>

                <String key="MaxTries"
                                value="20"
                                name="Max Tries"
                                description="Maximum number of tries to assign IP address
                                    per iteration."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
```

```
                        description="Hardware address of the network interface on ↩
                            machine running Peach Fuzzer. To find the hardware ↩
                            address on Windows, run 'ipconfig /all' and look for the ↩
                             'Physical Address' field. For Linux run 'ifconfig' and ↩
                            look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                             look for the 'ether' field."/>

        <Ipv6 key="SourceIPv6"
                  value="::1"
                  name="Source IPv6 Address"
                  description="The IPv6 address of the machine running Peach Fuzzer ↩
                      . The IPv6 address can be found on Windows by running ' ↩
                      ipconfig' and looking for the 'IPv6 Address' field. For Linux ↩
                      run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                      ifconfig' and look for the 'inet6' field."/>

        <Range key="SourcePort"
                   value="547"
                   min="0" max="65535"
                   name="Source Port"
                   description="The source port the network packet originates from. ↩
                       "/>

        <Hwaddr key="TargetMAC"
                        value="000000000000"
                        name="Target MAC Address"
                        description="Hardware address of the network interface on ↩
                            target machine or device. To find the hardware address ↩
                            on Windows, run 'ipconfig /all' and look for the ' ↩
                            Physical Address' field. For Linux run 'ifconfig' and ↩
                            look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                             look for the 'ether' field."  />

        <Ipv6 key="TargetIPv6"
                  value="::1"
                  name="Target IPv6 Address"
                  description="The IPv6 address of the target machine or device. ↩
                      The IPv6 address can be found on Windows by running 'ipconfig' ↩
                       and looking for the 'IPv6 Address' field. For Linux run ' ↩
                      ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                      ifconfig' and look for the 'inet6' field."/>

        <Range key="TargetPort"
                   value="546"
                   min="0"
                   max="65535"
                   name="Target Port"
                   description="The target or destination port the network packet ↩
                       is sent to."/>

        <String key="LoggerPath"
                        value="logs/dhcpv6_server/"
                        name="Logger Path"
                        description="The directory where Peach will save the log ↩
                            produced when fuzzing." />

        <Strategy key="Strategy"
                          value="Random"
                          name="Mutation Strategy"
                          description="The mutation strategy to use when fuzzing." ↩
                              />
```

```
                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 10.2** Sample DHCPv6 Server Configuration File

Example configuration sending DHCPv6 packets.
For this example we assume you are running on Windows.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)
```

NOTE: The MAC addresses must be updated based on the environment.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="TargetIPv6Lease"
                                value="fe80::908:7a38:6156:acce"
                                name="Target IPv6 Lease"
                                description="IPv6 address to assign."/>

                <String key="MaxTries"
                                value="20"
                                name="Max Tries"
                                description="Maximum number of tries to assign IP address ↩
                                    per iteration."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer ↩
                              . The IPv6 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv6 Address' field. For Linux ↩
                              run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="SourcePort"
                           value="547"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
```

```
                                description="Hardware address of the network interface on  ↩
                                    target machine or device. To find the hardware address  ↩
                                    on Windows, run 'ipconfig /all' and look for the ' ↩
                                    Physical Address' field. For Linux run 'ifconfig' and  ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."  />

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device.  ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="TargetPort"
                           value="546"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                                value="logs/dhcpv6_client/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 11 Ethernet

- Peach Pit: Ethernet
- Direction: Broadcast
- Supported Platforms: Linux

An Ethernet Frame is a link layer data packet as described in IEEE 802.3. It encapsulates all the layers above it.

This is a simple protocol; this pit would not be used alone, but would likely be used to construct encapsulated protocol pits.

## 11.1 Specifications

| Specification | Title |
|---|---|
| 802.3 | Ethernet |
| 802.1Q | VLAN Tagging |

## 11.2 Use Cases

| Messages | Specification |
|---|---|
| Ethernet Frame | 802.3 Section 3.1.1 |

| Supported Features | Specification |
|---|---|
| Address Fields | 802.3 Section 3.2.3 |
| Length/Type Fields (Partial) | 802.3 Section 3.2.6 |
| MAC Client Data field | 802.3 Section 3.2.7 |

## 11.3 Configuration

### 11.3.1 Target Configuration

This pit sends raw Ethernet frames; no extra applications are required.

### 11.3.2 Required Pit Configuration Changes

**TargetIPv4**
IPv4 address of the target host machine.

**SourceIPv4**
IPv4 address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface.

#### 11.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 11.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 11.4 Running

#### 11.4.1 Single test debug run

```
peach -1 --debug Ethernet.xml
```

#### 11.4.2 Full test run

```
peach Ethernet.xml
```

### 11.5 Examples

**Example 11.1** Sample Ethernet Configuration File

Example configuration for sending raw ethernet packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
```

```
                        name="Target MAC Address"
                        description="Hardware address of the network interface on ↩
                            target machine or device. To find the hardware address ↩
                            on Windows, run 'ipconfig /all' and look for the ' ↩
                            Physical Address' field. For Linux run 'ifconfig' and ↩
                            look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                             look for the 'ether' field."  />

        <Ipv4 key="TargetIPv4"
                  value="127.0.0.1"
                  name="Target IPv4 Address"
                  description="The IPv4 address of the target machine or device. ↩
                      The IPv4 address can be found on Windows by running 'ipconfig' ↩
                       and looking for the 'IPv4 Address' field. For Linux run ' ↩
                      ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                      ifconfig' and look for the 'inet' field." />

        <String key="LoggerPath"
                        value="logs/ethernet/"
                        name="Logger Path"
                        description="The directory where Peach will save the log ↩
                            produced when fuzzing." />

        <Strategy key="Strategy"
                          value="Random"
                          name="Mutation Strategy"
                          description="The mutation strategy to use when fuzzing." ↩
                              />

        <String key="PitLibraryPath"
                        value="."
                        name="Pit Library Path"
                        description="The path to the root of the pit library."/>
    </All>

    <Linux>
        <Iface key="Interface"
                  value="eth0"
                  name="Network Interface"
                  description="The network interface to transmit packets over. For ↩
                       Windows, the network interfaces can be shown by running ' ↩
                      ipconfig'. On Linux and OS X, the network interfaces can be ↩
                      shown by running the command 'ifconfig'."/>
    </Linux>

    <OSX>
        <Iface key="Interface"
                  value="en0"
                  name="Network Interface"
                  description="The network interface to transmit packets over. For ↩
                       Windows, the network interfaces can be shown by running ' ↩
                      ipconfig'. On Linux and OS X, the network interfaces can be ↩
                      shown by running the command 'ifconfig'."/>
    </OSX>

    <Windows>
        <Iface key="Interface"
                  value="Local Area Connection"
                  name="Network Interface"
                  description="The network interface to transmit packets over. For ↩
                       Windows, the network interfaces can be shown by running ' ↩
                      ipconfig'. On Linux and OS X, the network interfaces can be ↩
```

```
                                shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 12 File Transfer Protocol (FTP)

- Peach Pit: FTP
- Direction: Client, Server
- Supported Platforms: Windows, Linux, OS X

File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it.

## 12.1 Specifications

| Specification | Title |
|---|---|
| RFC 959 - File Transfer Protocol (FTP). | RFC 697 - CWD Command of FTP |

## 12.2 Use Cases

| Messages | Specification |
|---|---|
| FTP Protocol | RFC959 |
| CWD Command | RFC 697 |
| Directory Oriented Commands | RFC 775 |

## 12.3 Configuration

### 12.3.1 Target Configuration

An FTP server is required to test the client side of the FTP pit. On Linux, ftpd can be downloaded using apt-get.

### 12.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**SourcePort**
ftp port number of the local machine.

**TargetPort**
ftp port number of the target host machine.

**Interface**
Name of local interface (used for monitoring).

**FtpUser**
Username of the remote ftp account.

**FtpPass**
Password for the associated ftp user account.

**DataPort**
The port used to listen for incoming data connections.

**Program**
The client program to test when fuzzing as the server.

#### 12.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Agent**
Agent to run depending on the target OS. Do not change this value.

**Path**
Path to the relative base directory where all pits are located.

#### 12.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 12.4 Running

#### 12.4.1 Single test debug run

**Fuzzing an FTP client in active mode**

```
peach -1 --debug FTP_Client.xml
```

**Fuzzing an FTP client in passive mode**

```
peach -1 --debug FTP_Client_Passive.xml
```

**Fuzzing an FTP Server in active mode**

```
peach -1 --debug FTP_Server.xml
```

**Fuzzing an FTP Server in passive mode**

```
peach -1 --debug FTP_Server.xml Passive
```

#### 12.4.2 Full test run

**Fuzzing an FTP client in active mode**

```
peach FTP_Client.xml
```

**Fuzzing an FTP client in passive mode**

```
peach FTP_Client_Passive.xml
```

**Fuzzing an FTP server in active mode**

```
peach FTP_Server.xml
```

**Fuzzing an FTP server in passive mode**

```
peach FTP_Server_Passive.xml
```

## 12.5 Examples

**Example 12.1** Sample FTP Client Configuration File

Example configuration targeting the ftpd server on Linux.
First we must install ftpd; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for ftpd._

```
sudo apt-get install ftpd
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Range
                        key="CommandPort"
                        value="21"
                        min="0" max="65535"
                        name="FTP Command Port"
                        description="Port number for the main FTP communication. Typically  ↩
                            this is port 21." />

                <Range
                        key="DataPort"
                        value="31337"
                        min="0" max="65535"
                        name="FTP Data Port"
                        description="Port number to use for data channel. This is a  ↩
                            secondary port used to transfer file data and should not be port ↩
                             21." />

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range  key="AcceptTimeout"
                                value="5000"
                                min="0" max="10000000"
                                name="Timeout"
```

```
                                description="Timeout in milliseconds to wait for client  ↩
                                    connection. During fuzzing a timeout failure will cause  ↩
                                    the fuzzer to skip to the next iteration."/>

                <Range  key="Timeout"
                                value="5000"
                                min="0" max="10000000"
                                name="Timeout"
                                description="Timeout in milliseconds to wait for data to be ↩
                                     send or received. During fuzzing a timeout failure will ↩
                                     cause the fuzzer to skip to the next iteration."/>

                <String key="LoggerPath"
                                value="logs/ftp_server_passive/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 12.2** Sample FTP Server Configuration File

Example configuration targeting the ftpd server on Linux.
First we must install ftpd; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for ftpd._

```
sudo apt-get install ftpd
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FtpUser"
                                value="ftpuser"
                                name="FTP Username"
                                description="FTP username used with the FTP Password to  ↩
                                    authenticate to the FTP server being tested."/>
                <String key="FtpPass"
                                value="ftpuser123"
                                name="FTP Password"
                                description="FTP password used with the FTP Username to  ↩
                                    authenticate to the FTP server being tested."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>
```

```
                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="21"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <Range key="DataPort"
                           value="31337"
                           min="0" max="65535"
                           name="FTP Data Port"
                           description="Port number to use for data channel. This is a  ↩
                               secondary port used to transfer file data and should not be  ↩
                               port 21." />

                <Range  key="Timeout"
                                value="5000"
                                min="0" max="10000000"
                                name="Timeout"
                                description="Timeout in milliseconds to wait for data to be ↩
                                     send or received. During fuzzing a timeout failure will ↩
                                     cause the fuzzer to skip to the next iteration."/>

                <String key="LoggerPath"
                                value="logs/ftp_client/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 13 GIF Image Format

- Peach Pit: GIF
- Supported Platforms: Windows, Linux, OS X

Graphics Interchange Format (GIF) is an image format based on formerly patented compression techniques. It is commonly used for web images despite its licensing restraints.

GIF consists of a short header and collection of recursive, null-terminated blocks (which contain graphic settings, color tables, comments, and pixel data).

GIF uses LZW lossless encryption for packing and can support animated images.

## 13.1 Specifications

| Specification | Title |
|---|---|
| http://www.w3.org/Graphics/GIF/spec-gif87.txt | Graphics Interchange Format Version |
| http://www.w3.org/Graphics/GIF/spec-gif89a.txt | Graphics Interchange Format Version 89a |

## 13.2 Use Cases

| Supported Versions | Specification |
|---|---|
| GIF87 | Graphics Interchange Format Version |
| GIF89a | Graphics Interchange Format Version 89a |

| Supported Blocks | Specification |
|---|---|
| Header | Graphics Interchange Format Version 89a (17.0) |
| Logical Screen Descriptor | Graphics Interchange Format Version 89a (18.0) |
| Global Color Table | Graphics Interchange Format Version 89a (19.0) |
| Image Descriptor | Graphics Interchange Format Version 89a (20.0) |
| Local Color Table | Graphics Interchange Format Version 89a (21.0) |
| Table Based Image Data | Graphics Interchange Format Version 89a (22.0) |
| Graphic Control Extension | Graphics Interchange Format Version 89a (23.0) |
| Comment Extension | Graphics Interchange Format Version 89a (24.0) |
| Plain Text Extension | Graphics Interchange Format Version 89a (25.0) |
| Application Extension | Graphics Interchange Format Version 89a (26.0) |
| Trailer | Graphics Interchange Format Version 89a (27.0) |

## 13.3 Configuration

### 13.3.1 Target Configuration

You can fuzz a variety of image viewing programs (such as "feh" on Linux and "mspaint.exe" on Windows) with the GIF file format.

In the Gif.xml.config pit file, you can set the target program that you are fuzzing, such as feh or mspaint.exe. Normally you would set different target programs for different operating systems.

### 13.3.2 Required Pit Configuration Changes

**Seed**
Name of a valid GIF file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together with an asterisk wildcard.

**Target**
The program that will open the fuzzed GIF files

#### 13.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the GIF sample files are stored. Relative to Path.

#### 13.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 13.4 Running

#### 13.4.1 Single test debug run

```
peach -1 --debug Gif.xml
```

#### 13.4.2 Full test run

```
peach Gif.xml
```

### 13.5 Examples

**Example 13.1** Sample GIF Configuration File

Example configuration using feh on linux. The configuration file also contains settings for mspaint on Windows and preview on OSX:
First we must install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.gif"
                                name="Fuzzed Output File"
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="*.gif"
                                name="Seed File"
                                description="The name of the sample file to use when  ↩
                                    fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="LoggerPath"
                                value="./logs/gif/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use  ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see feh repeatedly open and close.

# 14 Internet Control Message Protocol version 4 (ICMPv4)

- Peach Pit: ICMPv4
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Internet Control Message Protocol version 4 (ICMPv4) is the implementation of a message relay protocol on top of IPv4. It provides error responses for the IPv4 protocol as well as simple data transmission. It is commonly used in the ping application for testing network response and connectivity.

## 14.1 Specifications

| Specification | Title |
|---|---|
| RFC792 | Internet Control Message Protocol |
| RFC1071 | Computing the Internet Checksum |
| RFC1122 | Requirements for Internet Hosts — Communication Layers |
| RFC1256 | ICMP Router Discovery Messages |
| RFC6918 | Formally Deprecating Some ICMPv4 Message Types |
| RFC6633 | Deprecation of ICMP Source Quench Messages |

## 14.2 Use Cases

| Messages | Specification |
|---|---|
| 8) Echo | RFC792 |
| 0) Echo Reply | RFC792 |
| 3) Destination Unreachable | RFC792 |
| 4) Source Quench | RFC792 |
| 5) Redirect | RFC792 |
| 11) Time Exceeded | RFC792 |
| 12) Parameter Problem | RFC792 |
| 13) Timestamp | RFC792 |
| 14) Timestamp Reply Problem | RFC792 |
| 15) Information Request | RFC792 |
| 16) Information Reply | RFC792 |
| 9) Router Advertisement | RFC1256 (Section 3) |
| 10) Router Solicitation | RFC1256 (Section 3) |

| Supported Features | Specification |
|---|---|
| Internet Checksum | RFC1071 |

| Supported Payloads | Specification |
|---|---|
| IPv4 in Error Responses | RFC1122 |

## 14.3 Configuration

### 14.3.1 Target Configuration

A target machine with ICMPv4 enabled. A firewall must not block ICMPv4 messages.

#### 14.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**Interface**
Name of local interface (used for monitoring).

#### 14.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 14.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 14.4 Running

#### 14.4.1 Single test debug run

```
peach -1 --debug ICMPv4.xml
```

#### 14.4.2 Full test run

```
peach ICMPv4.xml
```

### 14.5 Examples

**Example 14.1** Sample ICMPv4 Configuration File

Example configuration sending ICMPv4 packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Destination Unreachable Fragmentation Needed (ICMPv4-In)
File and Printer Sharing (Echo Request - ICMPv4-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
```

```
                        description="The IPv4 address of the machine running Peach Fuzzer ↩
                            . The IPv4 address can be found on Windows by running ' ↩
                            ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                            run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet' field."/>

                <Ipv4 key="TargetIPv4"
                        value="127.0.0.1"
                        name="Target IPv4 Address"
                        description="The IPv4 address of the target machine or device.  ↩
                            The IPv4 address can be found on Windows by running 'ipconfig' ↩
                             and looking for the 'IPv4 Address' field. For Linux run ' ↩
                            ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet' field." />

                <String key="LoggerPath"
                                value="logs/icmpv4/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 15 Internet Control Message Protocol version 6 (ICMPv6)

- Peach Pit: ICMPv6
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Internet Control Message Protocol version 6 (ICMPv6) is the implementation of the Internet Control Message Protocol (ICMP) for Internet Protocol version 6 (IPv6) defined in RFC 4443.

ICMPv6 is an integral part of IPv6 and performs error reporting, diagnostic functions (like ping), and a framework for extensions to implement future changes.

Several extensions have been published, defining new ICMPv6 message types as well as new options for existing ICMPv6 message types:

- Neighbor Discovery Protocol (NDP) is a node discovery protocol in IPv6 which replaces and enhances functions of ARP
- Secure Neighbor Discovery Protocol (SEND) is an extension of NDP with extra security
- Multicast Router Discovery (MRD) allows discovery of multicast routers

## 15.1 Specifications

| Specification | Title |
|---|---|
| RFC4443 | Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification |
| RFC2460 | Internet Protocol, Version 6 (IPv6) Specification |
| RFC2710 | Multicast Listener Discovery (MLD) for IPv6 |
| RFC4861 | Neighbor Discovery for IP version 6 (IPv6) |
| RFC6275 | Mobility Support in IPv6 |
| RFC4620 | IPv6 Node Information Queries |
| RFC3122 | Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification |
| RFC3810 | Multicast Listener Discovery Version 2 (MLDv2) for IPv6 |
| RFC4286 | Multicast Router Discovery |
| RFC4861 | Neighbor Discovery |
| RFC5568 | Mobile IPv6 Fast Handovers |
| RFC3971 | SEcure Neighbor Discovery (SEND) |
| RFC6550 | RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks |
| RFC6743 | ICMP Locator Update Message for the Identifier-Locator Network Protocol for IPv6 (ILNPv6) |

## 15.2 Use Cases

| Messages | Specification |
|---|---|
| Destination Unreachable | RFC4443 (Section 3.1) |
| Packet Too Big | RFC4443 (Section 3.2) |
| Time Exceeded | RFC4443 (Section 3.3) |
| Parameter Problem | RFC4443 (Section 3.4) |
| Echo Request | RFC4443 (Section 4.1) |
| Echo Reply | RFC4443 (Section 4.2) |

## 15.3 Configuration

### 15.3.1 Target Configuration

A target machine with ICMPv6 enabled. A firewall must not block ICMPv6 messages.

### 15.3.2 Required Pit Configuration Changes

**TargetIPv6**
IP address of the target host machine.

**SourceIPv6**
IP address of the interface on the local machine.

### 15.3.3 Optional Pit Configuration Changes.

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

### 15.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 15.4 Running

### 15.4.1 Single test debug run

```
peach -1 --debug ICMPv6.xml
```

### 15.4.2 Full test run

```
peach ICMPv6.xml
```

## 15.5 Examples

**Example 15.1** Sample ICMPv6 Configuration File

Example configuration sending ICMPv6 packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Destination Unreachable (ICMPv6-In)
Core Networking - Multicast Listener Done (ICMPv6-In)
Core Networking - Multicast Listener Query (ICMPv6-In)
Core Networking - Multicast Listener Report (ICMPv6-In)
Core Networking - Multicast Listener Report v2 (ICMPv6-In)
Core Networking - Neighbor Discovery Advertisement (ICMPv6-In)
Core Networking - Neighbor Discovery Solicitation (ICMPv6-In)
Core Networking - Packet Too Big (ICMPv6-In)
Core Networking - Parameter Problem (ICMPv6-In)
Core Networking - Router Advertisement (ICMPv6-In)
Core Networking - Router Solicitation (ICMPv6-In)
Core Networking - Time Exceeded (ICMPv6-In)
File and Printer Sharing (Echo Request - ICMPv6-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer ↩
                              . The IPv6 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv6 Address' field. For Linux ↩
                              run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device. ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <String key="LoggerPath"
                                value="logs/icmpv6/"
                                name="Logger Path"
                                description="The directory where Peach will save the log ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing." ↩
                                      />
```

```
                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 16 ICO Image Format

- Peach Pit: ICO
- Supported Platorms: Windows, Linux, OS X

The ICO file format is an image file format for computer icons in Microsoft Windows. ICO files contain one or more small images at multiple sizes and color depths, such that they may be scaled appropriately.

## 16.1 Specifications

| Specification | Title |
|---|---|
| http://msdn.microsoft.com/en-us/library/ms997538.aspx | Icons |

## 16.2 Configuration

### 16.2.1 Target Configuration

Any number of image viewing programs may be targeted with the Ico file format. Example targets for fuzzing Ico are "feh" on Linux and "mspaint.exe" on Windows.

In the Ico.xml.config pit file, you can set the target program that you are fuzzing, such as feh or mspaint.exe. Normally you would set different target programs for different operating systems.

### 16.2.2 Required Pit Configuration Changes

**Seed**
Name of a valid Ico file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together with an asterisk wildcard.

**Target**
The program that will open the fuzzed Ico files

### 16.2.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the ICO sample files are stored. Relative to Path.

### 16.2.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 16.3 Running

### 16.3.1 Single test debug run

```
peach -1 --debug Ico.xml
```

### 16.3.2 Full test run

```
peach Ico.xml
```

## 16.4 Examples

**Example 16.1** Sample ICO Configuration File

Example configuration using feh on linux. The configuration file also contains settings for mspaint on Windows and preview on OSX:
First we must install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.ico"
                                name="Fuzzed Output File"
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="*.ico"
                                name="Seed File"
                                description="The name of the sample file to use when  ↩
                                    fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="LoggerPath"
                                value="./logs/ico/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use  ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
```

```
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see feh repeatedly open and close.

# 17 Internet Group Management Protocol (IGMP)

- Peach Pit: IGMPv4
- Direction: Client
- Supported Platforms: Windows, Linux

Internet Group Management Protocol (IGMP) is used to declare and manage group membership for multicast addresses in IPv6.

Multicast IP Addresses allow machines to subscribe to IP address groups. Packets addressed to a group IP Address are distributed among all members.

IGMP is the protocol over IPv4 that allows machines to manage their group subscriptions.

## 17.1 Specifications

| Specification | Title |
|---|---|
| RFC3376 | Internet Group Management Protocol, Version 3 |
| RFC4604 | Using Internet Group Management Protocol Version 3 and Multicast Listener Discovery Protocol Version 2 for Source-Specific Multicast |
| RFC2236 | Internet Group Management Protocol, Version 2 |
| RFC1112 | Host Extensions for IP Multicasting |

## 17.2 Use Cases

| Messages | Specification |
|---|---|
| Membership Query | RFC3376 (Section 4.0) |
| Version 3 Membership Report | RFC3376 (Section 4.0) |

| Supported Features | Specification |
|---|---|
| General Query | RFC3376 (Section 4.1.11) |
| Group-Specific Query | RFC3376 (Section 4.1.11) |
| Group-and-Source-Specific Query | RFC3376 (Section 4.1.11) |

## 17.3 Configuration

### 17.3.1 Target Configuration

A target machine with IGMP enabled. A firewall must not block IGMP messages.

### 17.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine

**SourceIPv4**
IP address of the interface on the local machine

#### 17.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing

**LoggerPath**
Path to folder where logs will be stored

**Path**
Path to the relative base directory where all pits are located.

#### 17.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 17.4 Running

#### 17.4.1 Single test debug run

```
peach -1 --debug IGMP.xml
```

#### 17.4.2 Full test run

```
peach IGMP.xml
```

### 17.5 Examples

**Example 17.1** Sample IGMP Configuration File

Example configuration sending IGMP packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Internet Group Management Protocol (IGMP-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />
```

```
                <String key="LoggerPath"
                                value="logs/igmp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 18 Internet Protocol Security

- Peach Pit: IPsec
- Direction: Client
- Supported Platforms: Windows, Linux

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session.

IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and cryptographic key negotiations during the session. The protocol can be used to protect data flows between:

- a pair of hosts (host-to-host)
- a pair of security gateways (network-to-network)
- a security gateway and a host (network-to-host)

## 18.1 Specifications

| Specification | Title |
|---|---|
| RFC2403 | The Use of HMAC-MD5-96 within ESP and AH |
| RFC2404 | The Use of HMAC-SHA-1-96 within ESP and AH |
| RFC2405 | The ESP DES-CBC Cipher Algorithm With Explicit IV |
| RFC2410 | The NULL Encryption Algorithm and Its Use With IPsec |
| RFC2451 | The ESP CBC-Mode Cipher Algorithms |
| RFC2857 | The Use of HMAC-RIPEMD-160-96 within ESP and AH |
| RFC4302 | IP Authentication Header |
| RFC4303 | IP Encapsulating Security Payload |

## 18.2 Use Cases

| Messages | Specification |
|---|---|
| Encapsulating Security Payload Packet Format | RFC4303 2 |
| Authentication Header (AH) | RFC4302 |
| Transport Mode Processing | RFC4303 3.1.1 |
| Tunnel Mode Processing | RFC4303 3.1.2 |
| Separate Confidentiality and Integrity Algorithms | RFC4303 3.4.4.1 |
| ICV HMAC-MD5-96 | RFC2403 |
| ICV HMAC-SHA-1-96 | RFC2404 |
| ICV HMAC-RIPEMD-160-96 | RFC2857 |
| 3DES-CBC Cipher Encryption | RFC2405, RFC2451 |
| Null Encryption | RFC2410 |

## 18.3 Configuration

### 18.3.1 Target Configuration

An IPsec target configured for manual keying using the keys defined in the configuration file is required. IP-tools on Linux can be used.

Both a UDP and an TCP listener are required to run all the tests. The networking tool socat can be used as a listener.

#### 18.3.2 Required Pit Configuration Changes

**TargetIPv6**
IPv6 address of the target host machine.

**SourceIPv6**
IPv6 address of the interface on the local machine.

**TargetIPv4**
IPv4 address of the target host machine (used for encapsulating IPv4 in IPv6).

**SourceIPv4**
IPv4 address of the interface on the local machine (used for encapsulating IPv4 in IPv6).

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Mode**
Processing mode for IPsec can either be Tunnel or Transport.

**EncryptionAlg**
Encryption algorithm used when encrypting packets.

**CryptoKey**
Shared key used to encrypt packets.

**HashAlg**
Hashing algorithm used to provide data integrity.

**AuthKey**
Shared key used for HMAC hashing.

**IV**
Initialization vector used with the encryption algorithm.

**SPI**
Security parameter index used assigned to the local machine.

**SourcePort**
UDPv6 and/or TCPv6 port number of the local machine.

**TargetPort**
UDPv6 and/or TCPv6 port number of the target host machine.

#### 18.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 18.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 18.4 Running

### 18.4.1 Single test debug run

**Fuzzing IPSECv6 ESP**

```
peach -1 --debug IPSECv6_ESP.xml
```

**Fuzzing IPSECv6 AH**

```
peach -1 --debug IPSECv6_AH.xml
```

### 18.4.2 Full test run

**Fuzzing IPSECv6 ESP**

```
peach IPSECv6_ESP.xml
```

**Fuzzing IPSECv6 AH**

```
peach IPSECv6_AH.xml
```

## 18.5 Examples

**Example 18.1** Sample IPsec AH Configuration File

Example configuration using ipsec-tools on Linux.
First we must install ipsec-tools; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for ipsec-tools.

```
#This will enable to the target machine to both send and receive IPsec traffic from the  ↩
    specified IP addresses

sudo apt-get install ipsec-tools

#Create the following ipsec.config file changing the encryption keys, algorithms and IP  ↩
    address to match the pit config

## Start ipsec.config ##
flush;
spdflush;

add 10.0.1.1 10.0.1.2 ah 201 -A hmac-md5 "41414141414141414141414141414141414141414141";
add 10.0.1.2 10.0.1.1 ah 301 -A hmac-md5 "41414141414141414141414141414141414141414141";

spdadd 10.0.1.2 10.0.1.2 any -P out ipsec
    ah/transport//require;

spdadd 10.0.1.1 10.0.1.2 any -P in ipsec
    ah/transport//require;
## End ipsec.config ##

#Run setkey to enable changes

sudo setkey -f ipsec.config

#Setting up socat listener for UDP
```

```
sudo socat STDIO udp-listen:12345

#Setting up socat listener for TCP

sudo socat STDIO tcp-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Enum key="Mode"
                          value="Transport"
                          enumType="Peach.Enterprise.Pits.IpSecv6_Mode"
                          name="IPSEC Mode"
                          description="IPSEC mode of operation."/>

                <Enum key="HashAlg"
                          value="HMACMD5"
                          enumType="Peach.Enterprise.Pits.IpSecv6_HMAC"
                          name="HMAC Hash Algorithm"
                          description="The HMAC hash algorithm to use."/>

                <String key="AuthKey"
                                value="41414141414141414141414141414141414141414141"
                                name="HMAC Key"
                                description="HMAC authentication key. Length of this key is ↩
                                     dependent on the HMAC algorithm selected."/>
                <String key="SPI"
                                value="201"
                                name="Security Parameters Index (SPI)"
                                description="The SPI is an arbitrary 32-bit value that, in  ↩
                                    combination with the destination IP address and security ↩
                                     protocol (AH), uniquely identifies the Security  ↩
                                    Association for this datagram."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on  ↩
                                    machine running Peach Fuzzer. To find the hardware  ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and  ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                             ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                             run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                             ifconfig' and look for the 'inet' field."/>

                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer ↩
                              . The IPv6 address can be found on Windows by running ' ↩
                             ipconfig' and looking for the 'IPv6 Address' field. For Linux  ↩
                             run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                             ifconfig' and look for the 'inet6' field."/>
```

```
                <Range key="SourcePort"
                           value="1234"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
                                description="Hardware address of the network interface on  ↩
                                    target machine or device. To find the hardware address  ↩
                                    on Windows, run 'ipconfig /all' and look for the ' ↩
                                    Physical Address' field. For Linux run 'ifconfig' and  ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."  />

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device.  ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="TargetPort"
                           value="1234"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                                value="logs/ipsecv6_ah/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 18.2** Sample IPsec ESP Configuration File

Example configuration using ipsec-tools on Linux.
First we must install ipsec-tools; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for ipsec-tools.

```
#This will enable to the target machine to both send and receive IPsec traffic from the  ↩
    specified IP addresses

sudo apt-get install ipsec-tools

#Create the following ipsec.config file changing the encryption keys, algorithms and IP  ↩
    address to match the pit config

## Start ipsec.config ##
flush;
spdflush;

add 10.0.1.1 10.0.1.2 ah 201 -A hmac-md5 "41414141414141414141414141414141414141414141";
add 10.0.1.2 10.0.1.1 ah 301 -A hmac-md5 "41414141414141414141414141414141414141414141";

add 10.0.1.1 10.0.1.2 esp 201 -E 3des-cbc "aeaeaeaeaeaeaeaeae";
add 10.0.1.2 10.0.1.1 esp 301 -E 3des-cbc "aeaeaeaeaeaeaeaeae";

spdadd 10.0.1.2 10.0.1.2 any -P out ipsec
    esp/transport//require
    ah/transport//require;

spdadd 10.0.1.1 10.0.1.2 any -P in ipsec
    esp/transport//require
    ah/transport//require;
## End ipsec.config ##

#Run setkey to enable changes

sudo setkey -f ipsec.config

#Setting up socat listener for UDP

sudo socat STDIO udp-listen:12345

#Setting up socat listener for TCP

sudo socat STDIO tcp-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>

                <Enum key="Mode"
                          value="Transport"
                          enumType="Peach.Enterprise.Pits.IpSecv6_Mode"
                          name="IPSEC Mode"
                          description="IPSEC mode of operation."/>

                <Enum key="EncryptionAlg"
                          value="Aes128"
                          enumType="Peach.Enterprise.Pits.IpSecv6_Encryption"
                          name="Encryption Algorithm"
                          description="IPSEC encryption algorithm."/>

                <String key="CryptoKey"
```

```
                    value="4141414141414141414141414141414141"
                    name="Encryption Key"
                    description="Encryption key in HEX. For AES the key must be ↩
                        16 bytes long. For 3DES it must be 8 bytes long."/>
            <String key="IV"
                    value="baae9ef59ff1ee56211769bd91da50ed"
                    name="Initialization Vector (IV)"
                    description="Initialization vector (IV) in HEX. For AES the ↩
                        IV must be 16 bytes long. For 3DES is must be 8 bytes  ↩
                       long."/>

            <Enum key="HashAlg"
                  value="HMACMD5"
                  enumType="Peach.Enterprise.Pits.IpSecv6_HMAC"
                  name="HMAC Hash Algorithm"
                  description="The HMAC hash algorithm to use."/>

            <String key="AuthKey"
                    value="41414141414141414141414141414141414141414141"
                    name="HMAC Key"
                    description="HMAC authentication key. Length of this key is ↩
                        dependent on the HMAC algorithm selected."/>
            <String key="SPI"
                    value="201"
                    name="Security Parameters Index (SPI)"
                    description="The SPI is an arbitrary 32-bit value that, in  ↩
                       combination with the destination IP address and security ↩
                        protocol (AH), uniquely identifies the Security  ↩
                       Association for this datagram."/>

            <Hwaddr key="SourceMAC"
                    value="000000000000"
                    name="Source MAC Address"
                    description="Hardware address of the network interface on  ↩
                       machine running Peach Fuzzer. To find the hardware  ↩
                       address on Windows, run 'ipconfig /all' and look for the ↩
                        'Physical Address' field. For Linux run 'ifconfig' and  ↩
                       look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                        look for the 'ether' field."/>

            <Ipv4 key="SourceIPv4"
                  value="127.0.0.1"
                  name="Source IPv4 Address"
                  description="The IPv4 address of the machine running Peach Fuzzer ↩
                      . The IPv4 address can be found on Windows by running ' ↩
                      ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                      run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                      ifconfig' and look for the 'inet' field."/>

            <Ipv6 key="SourceIPv6"
                  value="::1"
                  name="Source IPv6 Address"
                  description="The IPv6 address of the machine running Peach Fuzzer ↩
                      . The IPv6 address can be found on Windows by running ' ↩
                      ipconfig' and looking for the 'IPv6 Address' field. For Linux  ↩
                      run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                      ifconfig' and look for the 'inet6' field."/>

            <Range key="SourcePort"
                   value="1234"
                   min="0" max="65535"
                   name="Source Port"
```

```
                        description="The source port the network packet originates from. ↩
                            "/>

                <Hwaddr key="TargetMAC"
                            value="000000000000"
                            name="Target MAC Address"
                            description="Hardware address of the network interface on  ↩
                                target machine or device. To find the hardware address  ↩
                                on Windows, run 'ipconfig /all' and look for the ' ↩
                                Physical Address' field. For Linux run 'ifconfig' and  ↩
                                look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                 look for the 'ether' field."  />

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device.  ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="TargetPort"
                           value="1234"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                            value="logs/ipsecv6_esp/"
                            name="Logger Path"
                            description="The directory where Peach will save the log  ↩
                                produced when fuzzing." />

                <Strategy key="Strategy"
                            value="Random"
                            name="Mutation Strategy"
                            description="The mutation strategy to use when fuzzing."  ↩
                                />

                <String key="PitLibraryPath"
                            value="."
                            name="Pit Library Path"
                            description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 19 Internet Protocol version 4 (IPv4)

- Peach Pit: IPv4
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Internet Protocol Version 4 (IPv4) is the Layer 3 Backbone of the internet. It is built on top of Ethernet and offers configurable addressing and routing capabilities to computer networks.

IPv4 is an inherently unreliable protocol because packets may not reach their destination. Layer 4 protocols like TCP are designed to provide reliable and ordered transmission of data over an IP network.

## 19.1 Specifications

| Specification | Title |
|---|---|
| RFC791 | INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION |
| RFC760 | DOD STANDARD INTERNET PROTOCOL |
| RFC1071 | Computing the Internet Checksum |
| RFC2474 | Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers |
| RFC3168 | The Addition of Explicit Congestion Notification (ECN) to IP |

## 19.2 Use Cases

| Messages | Specification |
|---|---|
| Packet | RFC791 |

| Supported Features | Specification |
|---|---|
| Internet Checksum | RFC1071 |
| Padding | RFC791 (Section 3.1) |
| Option Blocks | RFC791 (Section 3.1) |
| Explicit Congestion Notification | RFC3168 |

| Supported Payloads | Specification |
|---|---|
| ICMPv4 Echo Request | RFC792 |

## 19.3 Configuration

### 19.3.1 Target Configuration

A target machine with ICMPv4 enabled. A firewall must not block ICMPv4 messages.

### 19.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**Interface**
Name of local interface.

#### 19.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 19.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 19.4 Running

#### 19.4.1 Single test debug run

```
peach -1 --debug IPv4.xml
```

#### 19.4.2 Full test run

```
peach IPv4.xml
```

### 19.5 Examples

**Example 19.1** Sample IPv4 Configuration File

Example configuration sending IPv4 packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
File and Printer Sharing (Echo Request - ICMPv4-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Ipv4 key="TargetIPv4"
```

```
                        value="127.0.0.1"
                        name="Target IPv4 Address"
                        description="The IPv4 address of the target machine or device.  ↩
                            The IPv4 address can be found on Windows by running 'ipconfig' ↩
                             and looking for the 'IPv4 Address' field. For Linux run ' ↩
                            ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet' field." />

                <String key="LoggerPath"
                                value="logs/ipv4/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 20 Internet Protocol version 6 (IPv6)

- Peach Pit: IPv6
- Direction: Client
- Supported Platforms: Linux

Internet Protocol version 6 (IPv6) is the the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet.

IPv6 was developed to deal with the long-anticipated problem of IPv4 address exhaustion and eventually will replace IPv4.

## 20.1 Specifications

| Specification | Title |
|---|---|
| RFC2460 | Internet Protocol, Version 6 (IPv6) Specification |
| RFC2675 | IPv6 Jumbograms |
| RFC3775 | Mobility Support in IPv6 |

## 20.2 Use Cases

| Messages | Specification |
|---|---|
| IPv6 Header | RFC2460 3 |
| IPv6 Mobility Header | RFC3775 6.1.1 |
| IPv6 Mobility Binding Refresh Request Message | RFC 3775 6.1.2 |
| IPv6 Mobility Home Test Init Message | RFC 3775 6.1.3 |
| IPv6 Mobility Care-of Test Init Message | RFC 3775 6.1.4 |
| IPv6 Mobility Home Test Message | RFC 3775 6.1.5 |
| IPv6 Mobility Care-of Test Message | RFC 3775 6.1.6 |
| IPv6 Mobility Binding Update Message | RFC 3775 6.1.7 |
| IPv6 Mobility Binding Acknowledgment Message | RFC 3775 6.1.8 |
| IPv6 Mobility Binding Error Message | RFC 3775 6.1.9 |
| IPv6 Mobility Routing Type 2 Header | RFC 3775 6.4.1 |
| IPv6 Mobility ICMP Home Agent Address Discovery Request Message | RFC 3775 6.5 |
| IPv6 Mobility ICMP Home Agent Address Discovery Reply Message | RFC 3775 6.6 |
| IPv6 Mobility ICMP Mobile Prefix Solicitation Message Format | RFC 3775 6.7 |
| IPv6 Mobility ICMP Mobile Prefix Advertisement Message Format | RFC 3775 6.8 |
| IPv6 Header Hop-by-Hop Options | RFC2460 4.3 |
| IPv6 Header Routing Options | RFC2460 4.4 |
| IPv6 Header Fragment Options | RFC2460 4.5 |
| IPv6 Header Destination Options | RFC2460 4.6 |
| IPv6 Header Jumbo Payload Option | RFC2675 2 |
| IPv6 Mobility Binding Refresh Advice Option | RFC3775 6.2.4 |
| IPv6 Mobility Alternate Care-of Address Option | RFC3775 6.2.5 |
| IPv6 Mobility Nonce Indices Option | RFC3775 6.2.6 |
| IPv6 Mobility Home Address Option | RFC3775 6.3 |

## 20.3 Configuration

### 20.3.1 Target Configuration

IPv6 and UDPv6 listeners are required to run all tests. The network tool socat can be used as a listener.

### 20.3.2 Required Pit Configuration Changes

**TargetIPv6**
IP address of the target host machine.

**SourceIPv6**
IP address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface (used for monitoring).

**SourcePort**
UDPv6 port number of the local machine.

**TargetPort**
UDPv6 port number of the target host machine.

### 20.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

### 20.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 20.4 Running

### 20.4.1 Single test debug run

```
peach -1 --debug IPv6.xml
```

### 20.4.2 Full test run

```
peach IPv6.xml
```

## 20.5 Examples

**Example 20.1** Sample IPv6 Configuration File

Example configuration using socat on Linux.
First we must install and run socat; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat.

```
#Install socat
sudo apt-get install socat

# Used to listen for ipv6 traffic
socat STDIO ip-recv:ipv6

#Used to listen for UDPv6
socat STDIO udp6-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on machine running Peach Fuzzer. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."/>

                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer. The IPv6 address can be found on Windows by running 'ipconfig' and looking for the 'IPv6 Address' field. For Linux run 'ifconfig' and look for 'inet6 addr' field. For OS X run 'ifconfig' and look for the 'inet6' field."/>

                <Range key="SourcePort"
                           value="12345"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. "/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
                                description="Hardware address of the network interface on target machine or device. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."  />

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device. The IPv6 address can be found on Windows by running 'ipconfig' and looking for the 'IPv6 Address' field. For Linux run '
```

```
                            ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet6' field."/>

                <Range key="TargetPort"
                            value="12345"
                            min="0"
                            max="65535"
                            name="Target Port"
                            description="The target or destination port the network packet  ↩
                                is sent to."/>

                <String key="LoggerPath"
                                value="logs/vlan/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>

        <Linux>
                <Iface key="Interface"
                            value="eth0"
                            name="Network Interface"
                            description="The network interface to transmit packets over. For ↩
                                 Windows, the network interfaces can be shown by running ' ↩
                                ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                                shown by running the command 'ifconfig'."/>
        </Linux>

        <OSX>
                <Iface key="Interface"
                            value="en0"
                            name="Network Interface"
                            description="The network interface to transmit packets over. For ↩
                                 Windows, the network interfaces can be shown by running ' ↩
                                ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                                shown by running the command 'ifconfig'."/>
        </OSX>

        <Windows>
                <Iface key="Interface"
                            value="Local Area Connection"
                            name="Network Interface"
                            description="The network interface to transmit packets over. For ↩
                                 Windows, the network interfaces can be shown by running ' ↩
                                ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                                shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 21 JPEG2000 Image Format

- Peach Pit: JPEG2000
- Supported Platorms: Windows, Linux, OS X

JPEG 2000 is an Image File Format developed as the next evolution of the JPEG standard from 1992. The format supports bother lossless and lossy compression, multiple resolution representations, and better compression and error resilience than the previous standard. The file is structured as recursive "boxes" containing metadata and pixel streaming.

## 21.1 Specifications

| Specification | Title |
|---|---|
| ISO 15444 | JPEG 2000 image coding system |

## 21.2 Use Cases

| Supported Features | Specification |
|---|---|
| JPEG 2000 Signature Box | ISO 15444 (I.5.1) |
| File Type Box | ISO 15444 (I.5.2) |
| JP2 Header Box | ISO 15444 (I.5.3) |
| Image Header Box | ISO 15444 (I.5.3.1) |
| Bits Per Component Box | ISO 15444 (I.5.3.2) |
| Colour Specification Box | ISO 15444 (I.5.3.3) |
| Palette Box | ISO 15444 (I.5.3.4) |
| Component Mapping Box | ISO 15444 (I.5.3.5) |
| Channel Definition Box | ISO 15444 (I.5.3.6) |
| Resolution Box | ISO 15444 (I.5.3.7) |
| Capture Resolution Box | ISO 15444 (I.5.3.7.1) |
| Default Display Resolution Box | ISO 15444 (I.5.3.7.2) |
| Contiguous Codestream Box | ISO 15444 (I.5.4) |
| Intellectual Property Box | ISO 15444 (I.6) |
| XML Box | ISO 15444 (I.7.1) |
| UUID Box | ISO 15444 (I.7.2) |
| UUID Info Box | ISO 15444 (I.7.3) |
| UUID List Box | ISO 15444 (I.7.3.1) |
| URL Box | ISO 15444 (I.7.3.2) |

## 21.3 Configuration

### 21.3.1 Target Configuration

Any number of image viewing programs (like GIMP 2.8 on Linux and Windows) may be targeted with the JPEG2000 file format.

In the JPEG2000.xml.config pit file, you can set the target program (such as GIMP 2.8) that you are fuzzing . If available, you can set different target programs for different operating systems.

### 21.3.2 Required Pit Configuration Changes

**Seed**
: Name of a valid JPEG2000 file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together with an asterisk wildcard.

**Target**
The program that will open the fuzzed JPEG2000 files

#### 21.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the JPEG2000 sample files are stored. Relative to Path.

#### 21.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 21.4 Running

#### 21.4.1 Single test debug run

```
peach -1 --debug JPEG2000.xml
```

#### 21.4.2 Full test run

```
peach JPEG2000.xml
```

### 21.5 Examples

**Example 21.1** Sample JPEG2000 Configuration File

Example configuration using feh on Linux. The configuration file also contains settings for gimp on Windows and preview on OSX:
First we must install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.jp2"
                                name="Fuzzed Output File"
```

```
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="*.jp2"
                                name="Seed File"
                                description="The name of the sample file to use when  ↩
                                    fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="LoggerPath"
                                value="./logs/JPEG2000/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use  ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see feh repeatedly open and close.

# 22 JPG-JFIF Image Format

- Peach Pit: JPG-JFIF
- Supported Platorms: Windows, Linux, OS X

JPEG File Interchange Format (JFIF) is an image format built on a TLV segment structure. It is built to be an extensible format and supports a variety of compressions and encodings.

Even though its structure is very similar to the EXIF standard, these two formats are incompatible. Many programs ignore these differences in requirements.

## 22.1 Specifications

| Specification | Title |
|---|---|
| http://www.w3.org/Graphics/JPEG/jfif3.pdf | JPEG File Interchange Format |
| http://www.w3.org/Graphics/JPEG/itu-t81.pdf | Information Technology - Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines |

## 22.2 Use Cases

| Supported Features | Specification |
|---|---|
| Quantization | Digital Compression (B.2.4.1) |
| Huffman | Digital Compression (B.2.4.2) |
| Arithmetic Conditioning | Digital Compression (B.2.4.3) |
| Restart Interval | Digital Compression (B.2.4.4) |
| Comment | Digital Compression (B.2.4.5) |
| Application Data | Digital Compression (B.2.4.6) |

## 22.3 Configuration

### 22.3.1 Target Configuration

Any number of image viewing programs may be targeted with the JPG-JFIF file format. Example targets for fuzzing JPG-JFIF are "feh" on Linux and "mspaint.exe" on Windows.

In the jpg-jfif.xml.config pit file, you can set the target program that you are fuzzing, such as feh or mspaint.exe. Normally you would set different target programs for different operating systems.

### 22.3.2 Required Pit Configuration Changes

**Seed**
: Name of a valid JPG-JFIF file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together wtih an asterisk wildcard.

**Target**
: The program that will open the fuzzed JPG-JFIF files

### 22.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the JPG-JFIF sample files are stored. Relative to Path.

#### 22.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 22.4 Running

#### 22.4.1 Single test debug run

```
peach -1 --debug jpg-jfif.xml
```

#### 22.4.2 Full test run

```
peach jpg-jfif.xml
```

## 22.5 Examples

**Example 22.1** Sample JPG-JFIF Configuration File

Example configuration using feh on Linux. The configuration file also contains settings for mspaint on Windows and preview on OSX:
First we must install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on Linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.jpg"
                                name="Fuzzed Output File"
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="jpg-jfif*.jpg"
                                name="Seed File"
                                description="The name of the sample file to use when  ↩
                                    fuzzing." />
```

```
                <Strategy key="Strategy"
                                value="Random"
                                name="Mutation Strategy"
                                description="The mutation strategy to use when fuzzing."  ↩
                                    />

                <String key="LoggerPath"
                                value="./logs/jpg-jfif/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use  ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see feh repeatedly open and close.

# 23 Link Aggregation Control Protocol (LACP)

- Peach Pit: LACP
- Direction: Client
- Supported Platforms: Linux

Link aggregation is a computer networking term to describe various methods of combining (aggregating) multiple network connections in parallel to increase throughput beyond what a single connection could sustain, and to provide redundancy in case one of the links fail.

## 23.1 Specifications

| Specification | Title |
|---|---|
| IEEE 802.1AX | Link Aggregation (5.2) |

## 23.2 Use Cases

| Messages | Specification |
|---|---|
| Link Aggregation | IEE 802.1AX |

## 23.3 Configuration

### 23.3.1 Target Configuration

No extra applications are required for pit tests but we recommend a LACP enabled switch.

### 23.3.2 Required Pit Configuration Changes

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface.

**ActorPort**
Port number of the local machine.

**PartnerPort**
Port number of the target host machine.

### 23.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 23.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 23.4 Running

#### 23.4.1 Single test debug run

```
peach -1 -debug LACP.xml
```

#### 23.4.2 Full test run

```
peach LACP.xml
```

## 23.5 Examples

**Example 23.1** Sample LACP Configuration File

Example configuration for sending LACP packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Hwaddr key="ActorMAC"
                                value="0004961f506a"
                                name="Actor System ID"
                                description="The Actor's System ID, encoded as a MAC  ↩
                                    address."/>
                <Hwaddr key="PartnerMAC"
                                value="000000000000"
                                name="Partner System ID"
                                description="The Partner's System ID, encoded as a MAC  ↩
                                    address."/>
                <Range key="ActorPort"
                           value="18"
                           min="0" max="65535"
                           name="LACP Actor Port"
                           description="The port number assigned to the port by the Actor ( ↩
                               the System sending the PDU)."/>
                <Range key="PartnerPort"
                           value="0"
                           min="0" max="65535"
                           name="LACP Partner Port"
                           description="The port number associated with this link assigned  ↩
                               to the port by the Partner."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
```

```
                        description="Hardware address of the network interface on machine running Peach Fuzzer. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."/>

        <Hwaddr key="TargetMAC"
                        value="000000000000"
                        name="Target MAC Address"
                        description="Hardware address of the network interface on target machine or device. To find the hardware address on Windows, run 'ipconfig /all' and look for the ' Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."  />

        <String key="LoggerPath"
                        value="logs/lacp/"
                        name="Logger Path"
                        description="The directory where Peach will save the log produced when fuzzing." />

        <Strategy key="Strategy"
                          value="Random"
                          name="Mutation Strategy"
                          description="The mutation strategy to use when fuzzing."
                              />

        <String key="PitLibraryPath"
                        value="."
                        name="Pit Library Path"
                        description="The path to the root of the pit library."/>
    </All>

    <Linux>
            <Iface key="Interface"
                       value="eth0"
                       name="Network Interface"
                       description="The network interface to transmit packets over. For Windows, the network interfaces can be shown by running 'ipconfig'. On Linux and OS X, the network interfaces can be shown by running the command 'ifconfig'."/>
    </Linux>

    <OSX>
            <Iface key="Interface"
                       value="en0"
                       name="Network Interface"
                       description="The network interface to transmit packets over. For Windows, the network interfaces can be shown by running 'ipconfig'. On Linux and OS X, the network interfaces can be shown by running the command 'ifconfig'."/>
    </OSX>

    <Windows>
            <Iface key="Interface"
                       value="Local Area Connection"
                       name="Network Interface"
                       description="The network interface to transmit packets over. For Windows, the network interfaces can be shown by running 'ipconfig'. On Linux and OS X, the network interfaces can be
```

```
                                shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 24 Lightweight Directory Access Protocol (LDAP)

- Peach Pit: LDAP
- Direction: Client, Server
- Supported Platforms: Windows, Linux, OS X

Use the Lightweight Directory Access Protocol (LDAP) application protocol to access and maintain distributed directory information services over an Internet Protocol (IP) network.

## 24.1 Specifications

| Specification | Title |
|---|---|
| RFC4511 | Lightweight Directory Access Protocol (LDAP) : The Protocol |

## 24.2 Use Cases

| Messages | Specification |
|---|---|
| Bind Operation | RFC4511 4.2 |
| Unbind Operation | RFC4511 4.3 |
| Search Operation | RFC4511 4.5 |
| Add Operation | RFC4511 4.7 |
| Delete Operation | RFC4511 4.8 |
| Compare Opeartion | RFC4511 4.10 |
| Extended Operation | RFC4511 4.12 |

## 24.3 Configuration

### 24.3.1 Target Configuration

A configured LDAP server with an account that is defined in the configuration file is required.

OpenLDAP can be used on Linux.

### 24.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine

**SourceIPv4**
IP address of the interface on the local machine

**TargetPort**
LDAP port number of the target host machine

**Username**
Username used to login to the LDAP service

**Password**
Password for the user logging into the LDAP service

#### 24.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing

**LoggerPath**
Path to folder where logs will be stored

**Path**
Path to the relative base directory where all pits are located.

#### 24.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 24.4 Running

#### 24.4.1 Single test debug run

**Fuzzing an LDAP Client**

```
peach -1 --debug LDAP_Client.xml
```

**Fuzzing an LDAP Server**

```
peach -1 --debug LDAP_Server.xml
```

#### 24.4.2 Full test run

**Fuzzing an LDAP Client**

```
peach LDAP_Client.xml
```

**Fuzzing an LDAP Server**

```
peach LDAP_Server.xml
```

### 24.5 Examples

**Example 24.1** Sample LDAP Client Configuration File

Example configuration using OpenLdap on Linux.
First we must install OpenLDAP; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for OpenLDAP.

```
# Install OpenLdap client tools
sudo apt-get install ldap-utils
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="UserName"
                                value="cn=admin,dc=peach,dc=local"
                                name="LDAP Username"
```

```
                                description="Credentials to use when binding to LDAP server ↩
                                    . Example: cn=admin,dc=peach,dc=local."/>

                <String key="Password"
                                value="password"
                                name="LDAP Password"
                                description="Password to use for authenticating to LDAP  ↩
                                    server."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Range key="SourcePort"
                           value="389"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Range  key="Timeout"
                                value="5000"
                                min="0" max="10000000"
                                name="Timeout"
                                description="Timeout in milliseconds to wait for data to be ↩
                                     send or received. During fuzzing a timeout failure will ↩
                                     cause the fuzzer to skip to the next iteration."/>

                <String key="LoggerPath"
                                value="logs/vlan/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 24.2** Sample LDAP Server Configuration File

Example configuration using OpenLdap on Linux.
First we must install OpenLDAP; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for OpenLDAP.

```
#Install OpenLdap server
sudo apt-get install slapd ldap-utils migrationtools
```

```
#Configure slapd
dpkg-reconfigure slapd

  *Omit OpenLDAP server configuration? No
  *DNS domain name: peach.local
  *Name of your organization: Peach 3.0
  *Admin Password: password
  *Confirm Password: password
  *Database backend to use: BDB
  *Do you want your database to be removed when slapd is purged? No
  *Move old database? Yes
  *Allow LDAPv2 Protocol? No

#Edit migration tools common and edit the following parameters
pico /etc/migrate_comon.ph
  $DEFAULT_MAIL_DOMAIN = "peach.local";
  $DEFAULT_BASE = "dc=peach.local";
#Press ctrl+x to save

#Create a group and people ldif file
pico ~/grouppeople.ldif
  dn: ou=People,dc=peach,dc=local
  ou: People
  objectclass: organizationalUnit
#Press ctrl+x to save

#Use migration tools to export users on your system to ldif files
/usr/share/migrationtools/migrate_group.pl /etc/group ~/group.ldif
/usr/share/migrationtools/migrate_passwd.pl /etc/passwd ~/password.ldif

#Add the the created files to Ldap, the admin password is required after each of the  ↩
    following commands
cd ~
ldapadd -x -W -D "cn=admin,dc=peach,dc=local" -f grouppeople.ldif
ldapadd -x -W -D "cn=admin,dc=peach,dc=local" -f group.ldif
ldapadd -x -W -D "cn=admin,dc=peach,dc=local" -f password.ldif
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="UserName"
                                value="cn=admin,dc=peach,dc=local"
                                name="LDAP Username"
                                description="Credentials to use when binding to LDAP server ↩
                                    . Example: cn=admin,dc=peach,dc=local."/>

                <String key="Password"
                                value="password"
                                name="LDAP Password"
                                description="Password to use for authenticating to LDAP  ↩
                                    server."/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
```

```
                        value="389"
                        min="0"
                        max="65535"
                        name="Target Port"
                        description="The target or destination port the network packet ↩
                            is sent to."/>

                <Range  key="Timeout"
                            value="5000"
                            min="0" max="10000000"
                            name="Timeout"
                            description="Timeout in milliseconds to wait for data to be ↩
                                 send or received. During fuzzing a timeout failure will ↩
                                 cause the fuzzer to skip to the next iteration."/>

                <String key="LoggerPath"
                            value="logs/ldap/"
                            name="Logger Path"
                            description="The directory where Peach will save the log ↩
                                produced when fuzzing." />

                <Strategy key="Strategy"
                              value="Random"
                              name="Mutation Strategy"
                              description="The mutation strategy to use when fuzzing." ↩
                                  />

                <String key="PitLibraryPath"
                            value="."
                            name="Pit Library Path"
                            description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 25 Link Layer Discovery Protocol

- Peach Pit: LLDP
- Direction: Announce
- Supported Platforms: Linux

The Link Layer Discovery Protocol (LLDP) is a vendor-neutral protocol designed to allow advertisement of information about network infrastructure hardware by said hardware.

It is formally described in IEEE 802.1AB as *Station and Media Access Control Connectivity Discovery* and replaces proprietary protocols such as EDP, CDP, LLTD, and SONMP.

## 25.1 Specifications

| Specification | Title |
|---|---|
| IEEE 802.1AB | Station and Media Access Control Connectivity Discovery |

## 25.2 Use Cases

| Messages | Specification |
|---|---|
| LLDP Announce | IEE 802.1AB |

## 25.3 Configuration

### 25.3.1 Target Configuration

This pit broadcasts LLDP packets; no extra applications are required.

### 25.3.2 Required Pit Configuration Changes

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Interface**
Name of local interface.

**Hostname**
Host name of the switch.

### 25.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

### 25.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 25.4 Running

### 25.4.1 Single test debug run

```
peach -1 --debug LLDP.xml
```

### 25.4.2 Full test run

```
peach LLDP.xml
```

## 25.5 Examples

**Example 25.1** Sample LLDP Configuration File

Example configuration for sending LLDP packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="Hostname"
                                value="foobar"
                                name="Switch Hostname"
                                description="Hostname of the switch."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
                                description="Hardware address of the network interface on ↩
                                    target machine or device. To find the hardware address ↩
                                    on Windows, run 'ipconfig /all' and look for the ' ↩
                                    Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."  />

                <String key="LoggerPath"
                                value="logs/lldp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
```

```
                                description="The mutation strategy to use when fuzzing."  ↩
                                    />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>

        <Linux>
                <Iface key="Interface"
                           value="eth0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </Linux>

        <OSX>
                <Iface key="Interface"
                           value="en0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </OSX>

        <Windows>
                <Iface key="Interface"
                           value="Local Area Connection"
                           name="Network Interface"
                           description="The network interface to transmit packets over. For ↩
                                Windows, the network interfaces can be shown by running ' ↩
                               ipconfig'. On Linux and OS X, the network interfaces can be  ↩
                               shown by running the command 'ifconfig'."/>
        </Windows>
</PitDefines>
```

# 26 Multicast Listener Discovery Protocol (MLD)

- Peach Pit: MLD
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Multicast Listener Discovery Protocol (MLD) is used to declare and manage group membership for multicast addresses in IPv6.

The protocol is contained within IPv6 and its structure is similar to ICMPv6. It contains similar functionality to the Internet Group Management Protocol for IPv4.

Commands include adding and removing IP's from a group and querying servers for current group membership.

## 26.1 Specifications

| Specification | Title |
|---|---|
| RFC3810 | Multicast Listener Discovery Version 2 (MLDv2) for IPv6 |
| RFC4604 | Using Internet Group Management Protocol Version 3 and Multicast Listener Discovery Protocol Version 2 for Source-Specific Multicast |
| RFC2710 | Multicast Listener Discovery (MLD) for IPv6 |

## 26.2 Use Cases

| Messages | Specification |
|---|---|
| Multicast Listener Query Message | RFC3810 (Section 5.1) |
| Version 2 Multicast Listener Report Message | RFC3810 (Section 5.2) |

| Supported Features | Specification |
|---|---|
| General Query | RFC3810 (Section 5.1.13) |
| Multicast Address Specific Query | RFC3810 (Section 5.1.13) |
| Multicast Address and Source Specific Query | RFC3810 (Section 5.1.13) |

## 26.3 Configuration

### 26.3.1 Target Configuration

A target machine with MLD enabled. A firewall must not block MLD messages.

### 26.3.2 Required Pit Configuration Changes

**TargetIPv6**
IP address of the target host machine.

**SourceIPv6**
IP address of the interface on the local machine.

### 26.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 26.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 26.4 Running

#### 26.4.1 Single test debug run

```
peach -1 --debug MLD.xml
```

#### 26.4.2 Full test run

```
peach MLD.xml
```

## 26.5 Examples

**Example 26.1** Sample MLD Configuration File

Example configuration sending MLD packets.
For this example we assume you are running on Windows. For other platforms use the preferred way to configure the firewall.
Windows Firewall Inbound Rules to enable on Target:

```
Core Networking - Multicast Listener Done (ICMPv6-In)
Core Networking - Multicast Listener Query (ICMPv6-In)
Core Networking - Multicast Listener Report (ICMPv6-In)
Core Networking - Multicast Listener Report v2 (ICMPv6-In)
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer ↩
                              . The IPv6 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv6 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>
                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device.  ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>
```

```
                <String key="LoggerPath"
                                value="logs/mdl/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 27 Modbus (Modbus)

- Peach Pit: Modbus
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Modbus Protocol is a messaging structure developed by Modicon in 1979. It is used to establish master-slave/client-server communication between intelligent devices.

It is a de facto standard, truly open and the most widely used network protocol in the industrial manufacturing environment.

## 27.1 Specifications

| Specification | Title |
|---|---|
| http://www.modbus.org/docs/-Modbus_Application_Protocol_V1_1b.pdf | MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b |
| http://www.modbus.org/docs/-Modbus_over_serial_line_V1.pdf | MODBUS over serial line specification and implementation guide V1.0 |

## 27.2 Use Cases

| Messages | Specification |
|---|---|
| Modbus over TCP | http://www.modbus.org/docs/-Modbus_Application_Protocol_V1_1b.pdf |
| RTU Transmission Mode | http://www.modbus.org/docs/-Modbus_over_serial_line_V1.pdf (2.5.1) |
| ASCII Transmission Mode | http://www.modbus.org/docs/-Modbus_over_serial_line_V1.pdf (2.5.2) |

## 27.3 Configuration

### 27.3.1 Target Configuration

A Modbus server listening on the Modbus port defined in configuration file is required. The network tool socat can be used as the listener.

### 27.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**TargetPort**
Modbus port number of the target host machine.

**SerialPort**
The serial port of the local machine when using Modbus over serial.

**Baudrate**
The baud rate for the current serial port.

**Parity**
The parity bit value used for the current serial port.

**DataBits**
The number of data bits in each character for the current serial port.

**StopBits**
Number of bits sent at the end of every character for the current serial port.

**Handshake**
The current handshake protocol used by the current serial port.

#### 27.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Timeout**
How long to wait for incoming data.

**Path**
Path to the relative base directory where all pits are located.

#### 27.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 27.4 Running

#### 27.4.1 Single test debug run

**Fuzzing modbus via TCP**

```
peach modbus_tcp.xml
```

**Fuzzing modbus via ASCII Serial**

```
peach modbus_ascii_serial.xml
```

**Fuzzing modbus via RTU Serial**

```
peach modbus_rtu_serial.xml
```

#### 27.4.2 Full test run

**Fuzzing modbus via TCP**

```
peach modbus_tcp.xml
```

**Fuzzing modbus via ASCII Serial**

```
peach modbus_ascii_serial.xml
```

**Fuzzing modbus via RTU Serial**

```
peach modbus_rtu_serial.xml
```

## 27.5 Examples

**Example 27.1** Sample Modbus TCP Configuration File

Example configuration using socat on Linux.
First we must install and run socat; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat.

```
sudo apt-get install socat

socat tcp-l:502,fork exec:'/bin/cat'
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="502"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <Range  key="Timeout"
                                value="5000"
                                min="0" max="10000000"
                                name="Timeout"
                                description="Timeout in milliseconds to wait for data to be ↩
                                     send or received. During fuzzing a timeout failure will ↩
                                     cause the fuzzer to skip to the next iteration."/>


                <String key="LoggerPath"
                                value="logs/modbus_tcp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 28 Network Time Protocol (NTP)

- Peach Pit: NTP
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.

## 28.1 Specifications

| Specification | Title |
|---|---|
| RFC5905 | Network Time Protocol Version 4: Protocol and Algorithms Specification |

## 28.2 Use Cases

| Messages | Specification |
|---|---|
| List Peers | RFC5905 |
| Read List | RFC5905 |
| Monlist | RFC5905 |

## 28.3 Configuration

### 28.3.1 Target Configuration

A UDP listener listening on the UDPv4 port defined in configuration file is required. The network tool socat can be used as the listener.

### 28.3.2 Required Pit Configuration Changes

**TargetIPv4**
IPv4 address of the target host machine.

**SourcePort**
UDPv4 and/or TCPv4 port number of the local machine.

**TargetPort**
UDPv4 and/or TCPv4 port number of the target host machine.

### 28.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 28.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 28.4 Running

#### 28.4.1 Single test debug run

```
peach -1 --debug NTP.xml
```

#### 28.4.2 Full test run

```
peach NTP.xml
```

## 28.5 Examples

**Example 28.1** Sample NTP Configuration File

Example configuration using socat on Linux.
First we must install and run socat; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat.

```
sudo apt-get install socat

socat STDIO udp4-listen:123
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Range key="SourcePort"
                           value="123"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>


                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="123"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>


                <String key="LoggerPath"
```

```
                                value="logs/ntp/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 29 PNG Image Format

- Peach Pit: PNG
- Supported Platorms: Windows, Linux, OS X

The Portable Network Graphics (PNG) file format was defined to be a non-patented lossless image compression format for images in network communications. It supports RGB palettes, grayscale, and non-palette-bases RGB[A] images.

The file is composed of a magic number and a series of TLV chunks. Each chunk contains a type, length, data, and a checksum at the end.

## 29.1 Specifications

| Specification | Title |
|---|---|
| http://www.libpng.org/pub/png/spec/1.2/PNG-Contents.html | PNG (Portable Network Graphics) Specification, Version 1.2 |

## 29.2 Use Cases

| Supported Chunks | Specification |
|---|---|
| IHDR | PNG Spec v1.2 (4.1.1) |
| PLTE | PNG Spec v1.2 (4.1.2) |
| IDAT | PNG Spec v1.2 (4.1.3) |
| IEND | PNG Spec v1.2 (4.1.4) |
| tRNS | PNG Spec v1.2 (4.2.1.1) |
| gAMA | PNG Spec v1.2 (4.2.2.1) |
| cHRM | PNG Spec v1.2 (4.2.2.2) |
| sRGB | PNG Spec v1.2 (4.2.2.3) |
| iCCP | PNG Spec v1.2 (4.2.2.4) |
| tEXt | PNG Spec v1.2 (4.2.3.1) |
| zTXt | PNG Spec v1.2 (4.2.3.2) |
| iTXt | PNG Spec v1.2 (4.2.3.3) |
| bKGD | PNG Spec v1.2 (4.2.4.1) |
| pHYs | PNG Spec v1.2 (4.2.4.2) |
| sBIT | PNG Spec v1.2 (4.2.4.3) |
| sPLT | PNG Spec v1.2 (4.2.4.4) |
| hIST | PNG Spec v1.2 (4.2.4.5) |
| tIME | PNG Spec v1.2 (4.2.4.6) |
| oFFs | Unknown |
| cpIp | Unknown |

## 29.3 Configuration

### 29.3.1 Target Configuration

Any number of image viewing programs may be targeted with the PNG file format. Example targets for fuzzing PNG are "feh" on Linux and "mspaint.exe" on Windows.

In the Png.xml.config pit file, you can set the target program that you are fuzzing, such as feh or mspaint.exe. Normally you would set different target programs for different operating systems.

#### 29.3.2 Required Pit Configuration Changes

**Seed**
Name of a valid PNG file located in the Samples directory. An empty string indicates use all files in the directory. Multiple files can be globbed together with an asterisk wildcard.

**Target**
The program that will open the fuzzed PNG files.

#### 29.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Directory path to folder where logs will be stored.

**Agent**
The agent that will be ran to open the target program. This is generally OS dependent.

**Path**
Path to the relative base directory where all pits are located.

**SamplePath**
Path to the directory in which the PNG sample files are stored. Relative to Path.

#### 29.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 29.4 Running

#### 29.4.1 Single test debug run

```
peach -1 --debug Png.xml
```

#### 29.4.2 Full test run

```
peach Png.xml
```

### 29.5 Examples

**Example 29.1** Sample Png Configuration File

Example configuration using feh on Linux. The configuration file also contains settings for mspaint on Windows and preview on OSX:
First we must install feh; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for feh.

```
#Installing feh on linux
sudo apt-get install feh
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="FuzzedFile"
                                value="fuzzed.png"
                                name="Fuzzed Output File"
                                description="File that is generated by Peach when fuzzing.  ↩
                                    This file will be consumed by the target application." / ↩
                                    >

                <String key="Seed"
                                value="*.PNG"
                                name="Seed File"
                                description="The name of the sample file to use when  ↩
                                    fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="LoggerPath"
                                value="./logs/png/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <String key="SamplePath"
                                value="##PitLibraryPath##/_Common/Samples/Image"
                                name="Sample Path"
                                description="The directory containing the samples to use  ↩
                                    when fuzzing." />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library." />
        </All>
</PitDefines>
```

When running this you will see feh repeatedly open and close.

# 30 Simple Network Management Protocol Version 2c (SNMP)

- Peach Pit: SNMP
- Direction: Client, Server
- Supported Platforms: Windows, Linux, OS X

Simple Network Management Protocol (SNMP) is a protocol for network management. It is used for collecting information from, and configuring, network devices (such as servers, printers, hubs, switches, and routers) on an Internet Protocol (IP) network.

## 30.1 Specifications

| Specification | Title |
|---|---|
| RFC1901 | Introduction to Community-based SNMPv2 |
| RFC1907 | Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) |

## 30.2 Use Cases

| Messages | Specification |
|---|---|
| GetRequest | RFC1907 |
| GetNextRequest | RFC1907 |
| GetBulkRequest | RFC1907 |
| GetResponse | RFC1907 |

## 30.3 Configuration

### 30.3.1 Target Configuration

A UDP listener listening on the SNMP port defined in configuration file is required. The SNMP server snmpd on Linux can be used.

### 30.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**SourcePort**
SNMP port number of the local machine.

**TargetPort**
SNMP port number of the target host machine.

**CommString**
Community string used for by the SNMP server.

**Program**
The client program to test when fuzzing as the server.

#### 30.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Agent**
Agent to run depending on the target OS. Do not change this value.

**Path**
Path to the relative base directory where all pits are located.

#### 30.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 30.4 Running

#### 30.4.1 Single test debug run

**Fuzzing SNMP Client**

```
peach -1 --debug SNMP_Client.xml
```

**Fuzzing SNMP Server**

```
peach -1 --debug SNMP_Server.xml
```

#### 30.4.2 Full test run

**Fuzzing SNMP Client**

```
peach SNMP_Client.xml
```

**Fuzzing SNMP Server**

```
peach SNMP_Server.xml
```

### 30.5 Examples

**Example 30.1** Sample SNMP Client Configuration File

Example configuration using snmp on Linux.
**Install SNMP client**

```
sudo apt-get install snmp
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="CommString"
                                value="public"
                                name="Community String"
```

```
                                description="The SNMP community string to expect and  ↩
                                    respond to."/>

                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                             ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                             run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                             ifconfig' and look for the 'inet' field."/>

                <Range key="SourcePort"
                           value="162"
                           min="0" max="65535"
                           name="Source Port"
                           description="Port number to listen for incoming packets on."/>

                <String key="LoggerPath"
                                value="logs/snmp_server/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 30.2** Sample SNMP Server Configuration File

Example configuration using snmpd on Linux.
First we must install snmpd; for this example we assume you are running on Ubuntu or Debian. For other platforms follow the platform specific installation instructions for snmpd.

```
# Install SNMP daemon
sudo apt-get install snmpd

# Add the IP address you want snmpd to listen on into the configuration file
sudo pico /etc/default/snmpd
  #Add the IP address of the interface the service will be listening on to the end of this  ↩
      line
  SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux,mteTrigger,mteTriggerConf -p /var/ ↩
      run/snmpd.pid'
  #IP address has been added to the end of the line
  SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux,mteTrigger,mteTriggerConf -p /var/ ↩
      run/snmpd.pid 192.168.222.130'
  #Save and exit

# Restart snmpd
sudo /etc/init.d/snmpd restart
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<PitDefines>
        <All>
                <String key="CommString"
                                value="public"
                                name="SNMP Community String"
                                description="SNNP community string to use in requests. The  ↩
                                    target SNMP server must be configured to respond to this ↩
                                     community string."/>

                <Range key="SourcePort"
                           value="161"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="162"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                                value="logs/snmp_client/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 31 Transmission Control Protocol Version 4 (TCPv4)

- Peach Pit: TCPv4
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP.

TCP provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to a local area network, Intranet or the public Internet.

## 31.1 Specifications

| Specification | Title |
|---|---|
| RFC793 | Transmission Control Protocol |

## 31.2 Use Cases

| Messages | Specification |
|---|---|
| MaximumSegmentSize | RFC793 Section 3.1 page 17 |

| Supported Features | Specification |
|---|---|
| Establishing a connection (Async, Client) | RFC793 Section 3.4 |
| Closing a Connection (Async) | RFC793 Section 3.5 Case 2 |
| Data Communication | RFC793 Section 3.7 |

## 31.3 Configuration

### 31.3.1 Target Configuration

A TCP listener listening on the TCPv4 port defined in configuration file is required. The network tool socat can be used as the listener.

To use this pit, disable outgoing RST packets; Peach manages TCP states outside of the kernel context.

### 31.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**TargetIPBytes**
IP address of the target host machine in hexadecimal.

**SourceIPBytes**
IP address of the interface on the local machine in hexadecimal.

**SourcePort**
TCPv4 port number of the local machine.

**TargetPort**
TCPv4 port number of the target host machine.

#### 31.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 31.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 31.4 Running

#### 31.4.1 Single test debug run

```
peach -1 --debug TCPv4.xml
```

#### 31.4.2 Full test run

```
peach TCPv4.xml
```

### 31.5 Examples

**Example 31.1** Sample TCPv4 Configuration File

Example configuration using socat on Linux.
First we must configure the firewall then install and run socat; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat and firewall configuration.

```
#Disable outgoing RST on the node running Peach
sudo iptables -A OUTPUT -p tcp -m tcp --tcp-flags RST RST -j DROP

#Install socat

sudo apt-get install socat

#Set up TCPv4 listener
socat tcp4-l:12345,fork,reuseaddr STDIO
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                             ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                             run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                             ifconfig' and look for the 'inet' field."/>

                <Range key="SourcePort"
                           value="1234"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                             ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                             ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="12345"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>


                <String key="LoggerPath"
                                value="logs/tcpv4/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 32 Transmission Control Protocol Version 6 (TCPv6)

- Peach Pit: TCPv6
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to a local area network, Intranet or the public Internet.

## 32.1 Specifications

| Specification | Title |
|---|---|
| RFC793 | Transmission Control Protocol |

## 32.2 Use Cases

| Messages | Specification |
|---|---|
| MaximumSegmentSize | RFC793 Section 3.1 page 17 |

| Supported Features | Specification |
|---|---|
| Establishing a connection (Async, Client) | RFC793 Section 3.4 |
| Closing a Connection (Async) | RFC793 Section 3.5 Case 2 |
| Data Communication | RFC793 Section 3.7 |

## 32.3 Configuration

### 32.3.1 Target Configuration

An application must be listening on the TCP port defined in the configuration file. The network tool socat can be used as the listener.

Disable outgoing RST packets when using this pit because Peach manages TCP state outside of the kernel context.

### 32.3.2 Required Pit Configuration Changes

**TargetIPv6**
IP address of the target host machine.

**SourceIPv6**
IP address of the interface on the local machine.

**SourcePort**
TCPv6 port number of the local machine.

**TargetPort**
TCPv6 port number of the target host machine.

#### 32.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 32.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 32.4 Running

#### 32.4.1 Single test debug run

```
peach -1 --debug TCPv6.xml
```

#### 32.4.2 Full test run

```
peach TCPv6.xml
```

### 32.5 Examples

**Example 32.1** Sample TCPv6 Configuration File

Example configuration using socat on Linux.
First we must configure the firewall then install and run socat; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat and firewall configuration.

```
# Disable outgoing RST on the node running Peach
ip6tables -A OUTPUT -p tcp -m tcp --tcp-flags RST RST -j DROP

# Add a local IPv6 interface, this becomes your source ip
ip addr add ::2 dev lo

# Install socat

sudo apt-get install socat

# Set up TCPv6 listener
socat tcp6-l:4321,fork,reuseaddr STDIO
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
```

```
                        description="The IPv6 address of the machine running Peach Fuzzer ↩
                            . The IPv6 address can be found on Windows by running ' ↩
                            ipconfig' and looking for the 'IPv6 Address' field. For Linux  ↩
                            run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet6' field."/>

            <Range key="SourcePort"
                       value="1234"
                       min="0" max="65535"
                       name="Source Port"
                       description="The source port the network packet originates from. ↩
                           "/>

            <Ipv6 key="TargetIPv6"
                      value="::1"
                      name="Target IPv6 Address"
                      description="The IPv6 address of the target machine or device.  ↩
                          The IPv6 address can be found on Windows by running 'ipconfig' ↩
                           and looking for the 'IPv6 Address' field. For Linux run ' ↩
                          ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                          ifconfig' and look for the 'inet6' field."/>

            <Range key="TargetPort"
                       value="4321"
                       min="0"
                       max="65535"
                       name="Target Port"
                       description="The target or destination port the network packet  ↩
                           is sent to."/>

            <String key="LoggerPath"
                            value="logs/tcpv6/"
                            name="Logger Path"
                            description="The directory where Peach will save the log  ↩
                                produced when fuzzing." />

            <Strategy key="Strategy"
                              value="Random"
                              name="Mutation Strategy"
                              description="The mutation strategy to use when fuzzing."  ↩
                                  />

            <String key="PitLibraryPath"
                            value="."
                            name="Pit Library Path"
                            description="The path to the root of the pit library."/>
    </All>
```

# 33 Telnet

- Peach Pit: Telnet
- Direction: Client, Server
- Supported Platforms: Windows, Linux, OS X

Telnet is a network protocol used on the Internet or on local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP).

## 33.1 Specifications

| Specification | Title |
|---|---|
| RFC137 | Telnet protocol specification |
| RFC139 | Telnet protocol specification |
| RFC854 | Telnet protocol specification |
| RFC855 | Telnet option specifications |
| RFC857 | Telnet echo option |
| RFC858 | Telnet suppress go ahead option |
| RFC859 | Telnet status option |
| RFC860 | Telnet timing mark option |
| RFC885 | Telnet end of record option |
| RFC1041 | Telnet 3270 regime option |
| RFC1073 | Telnet window size option |
| RFC1079 | Telnet terminal speed option |
| RFC1091 | Telnet terminal-type option |
| RFC1096 | Telnet X display location option |
| RFC1116 | Telnet Linemode option |
| RFC1184 | Telnet Linemode option |
| RFC1372 | Telnet remote flow control option |
| RFC1572 | Telnet environment option |

## 33.2 Use Cases

| Messages | Specification |
|---|---|
| Telnet Protocol | RFC137 |
| Sub Options | RFC855 |
| Echo Option | RFC 857 |
| SGA Option | RFC858 |
| Status Option | RFC859 |
| Timing Mark Option | RFC860 |
| End of Record Option | RFC885 |
| Window Size Option | RFC1073 |
| Terminal Speed Option | RFC1079 |
| Terminal-type option | RFC1091 |
| X Display Option | RFC1096 |
| Linemode Option | RFC1116 |
| Remote Flow Control Option | RFC1372 |
| Environment Option | RFC1572 |

## 33.3 Configuration

### 33.3.1 Target Configuration

A Telnet server is required to test the client side of the Telnet pit. On Linux, telnetd can be downloaded using apt-get.

### 33.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**SourcePort**
Telnet port number of the local machine.

**TargetPort**
Telnet port number of the target host machine.

**Interface**
Name of local interface (used for monitoring).

**Username**
User name of the remote Telnet account.

**Password**
Password for the associated Telnet user account.

**Program**
The client program to test when fuzzing as the server.

### 33.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

### 33.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 33.4 Running

### 33.4.1 Single test debug run

**Fuzzing a telnet client**

```
peach -1 --debug TELNET_Client.xml
```

**Fuzzing a telnet server**

```
peach -1 --debug TELNET_Server.xml
```

### 33.4.2 Full test run

**Fuzzing a telnet client**

```
peach TELNET_Client.xml
```

**Fuzzing a telnet server**

```
peach TELNET_Server.xml
```

## 33.5 Examples

**Example 33.1** Sample Telnet Server Configuration File

Example configuration using telnetd on Linux.
First we must install the telnet server telnetd; for this example we assume you are running on Ubuntu or Debian. A test user account must also be created.

```
# Install telnet server on Linux
sudo apt-get install telnetd

# Add a test user account with password telnetuserp455
sudo adduser telnetuser
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <String key="Username"
                                value="telnetuser"
                                name="Username"
                                description="Username to use when logging into telnet  ↩
                                    server."/>
                <String key="Password"
                                value="telnetuserp455"
                                name="Password"
                                description="Password to use when logging into telnet  ↩
                                    server."/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="23"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                                value="logs/telnet_client/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />
```

```
                <Strategy key="Strategy"
                                value="Random"
                                name="Mutation Strategy"
                                description="The mutation strategy to use when fuzzing."  ↩
                                    />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

**Example 33.2** Sample Telnet Client Configuration File

The Telnet Client Pit file will simulate a telnet server to perform fuzzing of a telnet client application. This example configuration assumes the user is running on Ubuntu Server LTS.

**Install telnet on Linux**

```
sudo apt-get install telnet
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Range key="SourcePort"
                           value="23"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <String key="LoggerPath"
                                value="logs/telnet_server/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                value="Random"
                                name="Mutation Strategy"
                                description="The mutation strategy to use when fuzzing."  ↩
                                    />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 34 User Datagram Protocol version 4 (UDPv4)

- Peach Pit: UDPv4
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite. It allows applications to send messages (called datagrams) to other hosts on an Internet Protocol (IP) network. Messages are stateless and can be sent without prior communications to set up special transmission channels or data paths.

UDP uses a simple transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any underlying network protocol unreliability to the user's program. As this is normally IP over unreliable media, there is no guarantee of delivery, ordering, or duplicate protection.

## 34.1 Specifications

| Specification | Title |
|---|---|
| RFC768 | User Datagram Protocol |

## 34.2 Use Cases

| Messages | Specification |
|---|---|
| UDP Header | RFC768 |

## 34.3 Configuration

### 34.3.1 Target Configuration

A UDP listener listening on the UDPv4 port defined in configuration file is required. The network tool socat can be used as the listener.

### 34.3.2 Required Pit Configuration Changes

**TargetIPv4**
IP address of the target host machine.

**SourceIPv4**
IP address of the interface on the local machine.

**SourcePort**
UDPv4 port number of the local machine.

**TargetPort**
UDPv4 port number of the target host machine.

### 34.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 34.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 34.4 Running

#### 34.4.1 Single test debug run

```
peach -1 --debug UDPv4.xml
```

#### 34.4.2 Full test run

```
peach UDPv4.xml
```

## 34.5 Examples

**Example 34.1** Sample UDPv4 Configuration File

Example configuration using socat on Linux.
First we must install socat and configure socat to listen for packets; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat.

```
sudo apt-get install socat

socat STDIO udp4-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <All>
                <Ipv4 key="SourceIPv4"
                          value="127.0.0.1"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux  ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>

                <Range key="SourcePort"
                           value="1234"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Ipv4 key="TargetIPv4"
                          value="127.0.0.1"
                          name="Target IPv4 Address"
```

```
                        description="The IPv4 address of the target machine or device.  ↩
                            The IPv4 address can be found on Windows by running 'ipconfig' ↩
                             and looking for the 'IPv4 Address' field. For Linux run ' ↩
                            ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet' field." />

                <Range key="TargetPort"
                           value="12345"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <String key="LoggerPath"
                                value="logs/udpv4/"
                                name="Logger Path"
                                description="The directory where Peach will save the log  ↩
                                    produced when fuzzing." />

                <Strategy key="Strategy"
                                  value="Random"
                                  name="Mutation Strategy"
                                  description="The mutation strategy to use when fuzzing."  ↩
                                      />

                <String key="PitLibraryPath"
                                value="."
                                name="Pit Library Path"
                                description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 35 User Datagram Protocol version 6 (UDPv6)

- Peach Pit: UDPv6
- Direction: Client
- Supported Platforms: Windows, Linux, OS X

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite, which allows computer applications to send messages (which may be referred to as datagrams) to other hosts on an Internet Protocol (IP) network without prior communications in order to set up special transmission channels or data paths.

UDP uses a simple transmission model with a minimum of protocol mechanism. UDP is a protocol over IP; it does not add any complexity to verify that its messages got to where they were supposed to go.

UDP has no handshaking dialogs; the user's program is exposed to any underlying network protocol unreliability. This means that there is no guarantee of delivery, ordering, or duplicate protection.

## 35.1 Specifications

| Specification | Title |
|---|---|
| RFC768 | User Datagram Protocol |

## 35.2 Use Cases

| Messages | Specification |
|---|---|
| UDP Header | RFC768 |

## 35.3 Configuration

### 35.3.1 Target Configuration

A UDP listener listening on the UDPv6 port defined in configuration file is required. The network tool socat can be used as the listener.

### 35.3.2 Required Pit Configuration Changes

**TargetIPv6**
IP address of the target host machine.

**SourceIPv6**
IP address of the interface on the local machine.

**SourcePort**
UDPv6 port number of the local machine.

**TargetPort**
UDPv6 port number of the target host machine.

### 35.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 35.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

## 35.4 Running

#### 35.4.1 Single test debug run

```
peach -1 --debug UDPv6.xml
```

#### 35.4.2 Full test run

```
peach UDPv6.xml
```

## 35.5 Examples

**Example 35.1** Sample UDPv6 Configuration File

Example configuration using socat on Linux.
First we must install socat and configure socat to listen for packets; for this example we assume you are running Linux. For other platforms follow the platform specific installation instructions for socat.

```
sudo apt-get install socat

socat STDIO udp6-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
  <All>
          <Ipv6 key="SourceIPv6"
                        value="::1"
                        name="Source IPv6 Address"
                        description="The IPv6 address of the machine running Peach Fuzzer.  ↩
                            The IPv6 address can be found on Windows by running 'ipconfig'  ↩
                            and looking for the 'IPv6 Address' field. For Linux run ' ↩
                            ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                            ifconfig' and look for the 'inet6' field."/>

          <Range key="SourcePort"
                         value="1234"
                         min="0" max="65535"
                         name="Source Port"
                         description="The source port the network packet originates from."/ ↩
                             >

          <Ipv6 key="TargetIPv6"
                        value="::1"
                        name="Target IPv6 Address"
```

```
                        description="The IPv6 address of the target machine or device. The  ↩
                            IPv6 address can be found on Windows by running 'ipconfig' and  ↩
                            looking for the 'IPv6 Address' field. For Linux run 'ifconfig'  ↩
                            and look for 'inet6 addr' field. For OS X run 'ifconfig' and  ↩
                            look for the 'inet6' field."/>

        <Range key="TargetPort"
                        value="12345"
                        min="0"
                        max="65535"
                        name="Target Port"
                        description="The target or destination port the network packet is  ↩
                            sent to."/>


        <String
                key="LoggerPath"
                value="logs/udpv6/"
                name="Logger Path"
                description="The directory where Peach will save the log produced when  ↩
                    fuzzing." />

        <Strategy
                key="Strategy"
                value="Random"
                name="Mutation Strategy"
                description="The mutation strategy to use when fuzzing." />

        <String
                key="PitLibraryPath"
                value="."
                name="Pit Library Path"
                description="The path to the root of the pit library."/>
  </All>
</PitDefines>
```

# 36 Virtual Local Area Network (VLAN)

- Peach Pit: VLAN
- Direction: Client
- Supported Platforms: Linux

In computer networking, a single layer-2 network may be partitioned to create multiple, mutually isolated, distinct broadcast domains, so that packets can only pass between them via one or more routers. Such a domain is referred to as a virtual local area network, virtual LAN or VLAN.

## 36.1 Specifications

| Specification | Title |
|---|---|
| IEEE 802.1Q | Media Access Control Bridges and Virtual Bridge Local Area Networks |
| IEEE 802.1ad | Virtual Bridged Local Area Networks Amendment 4: Provider Bridges |

## 36.2 Use Cases

| Messages | Specification |
|---|---|
| VLAN Header | IEEE 802.1Q |
| VLAN Double Tagging | IEEE 802.1ad |

## 36.3 Configuration

### 36.3.1 Target Configuration

A VLAN interface or VLAN enabled switch or other device is required.

In order to run all tests both a UDP and TCP listener are required. The networking tool `socat` can be used as the listener.

### 36.3.2 Required Pit Configuration Changes

**TargetIPv6**
IPv6 address of the target host machine.

**SourceIPv6**
IPv6 address of the interface on the local machine.

**TargetIPv4**
IPv4 address of the target host machine.

**SourceIPv4**
IPv4 address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**Tag**
VLAN identifier specifying the VLAN to which the frames belong.

**SourcePort**
UDPv6 and/or TCPv6 port number of the local machine.

**TargetPort**
UDPv6 and/or TCPv6 port number of the target host machine.

#### 36.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 36.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 36.4 Running

#### 36.4.1 Single test debug run

```
peach -1 --debug VLAN.xml
```

#### 36.4.2 Full test run

```
peach VLAN.xml
```

### 36.5 Examples

**Example 36.1** Sample VLAN Configuration File

In this example, a Linux machine is configured with vlan support. We recommend using the current release of Ubuntu Server LTS. In order to replicate this example, first install the `vlan` package and configure. Then, set up `socat` in different terminals to listen for incoming packets.

`NOTE: The TargetMAC configuration parameter must be correctly configured.`

```
# Install vlan
sudo apt-get install vlan

# Add a new vlan interface with the tag of 1
sudo vconfig add eth0 1

# Config the vlan interface
sudo ifconfig eth0.1 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255 up

# Install socat
```

```
sudo apt-get install socat

# Setting up socat listener for UDP
sudo socat STDIO udp-listen:12345

# Setting up socat listener for TCP
sudo socat STDIO tcp-listen:12345
```

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
        <!-- NOTE: This pit is only supported on Linux or OS X. -->
        <All>
                <Iface key="Interface"
                           value="eth0"
                           name="Network Interface"
                           description="The network interface to transmit packets over. On ↩
                               Linux and OS X, the network interfaces can be shown by ↩
                               running the command 'ifconfig'."/>

                <Hwaddr key="SourceMAC"
                                value="000000000000"
                                name="Source MAC Address"
                                description="Hardware address of the network interface on ↩
                                    machine running Peach Fuzzer. To find the hardware ↩
                                    address on Windows, run 'ipconfig /all' and look for the ↩
                                     'Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."/>

                <Ipv4 key="SourceIPv4"
                          value="192.168.1.2"
                          name="Source IPv4 Address"
                          description="The IPv4 address of the machine running Peach Fuzzer ↩
                              . The IPv4 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv4 Address' field. For Linux ↩
                              run 'ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field."/>
                <Ipv6 key="SourceIPv6"
                          value="::1"
                          name="Source IPv6 Address"
                          description="The IPv6 address of the machine running Peach Fuzzer ↩
                              . The IPv6 address can be found on Windows by running ' ↩
                              ipconfig' and looking for the 'IPv6 Address' field. For Linux ↩
                              run 'ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="SourcePort"
                           value="1234"
                           min="0" max="65535"
                           name="Source Port"
                           description="The source port the network packet originates from. ↩
                               "/>

                <Hwaddr key="TargetMAC"
                                value="000000000000"
                                name="Target MAC Address"
                                description="Hardware address of the network interface on ↩
                                    target machine or device. To find the hardware address ↩
                                    on Windows, run 'ipconfig /all' and look for the ' ↩
                                    Physical Address' field. For Linux run 'ifconfig' and ↩
                                    look for the 'HWaddr' field. For OS X run 'ifconfig' and ↩
                                     look for the 'ether' field."  />
```

```
                <Ipv4 key="TargetIPv4"
                          value="192.168.1.1"
                          name="Target IPv4 Address"
                          description="The IPv4 address of the target machine or device.  ↩
                              The IPv4 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv4 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet' field." />

                <Ipv6 key="TargetIPv6"
                          value="::1"
                          name="Target IPv6 Address"
                          description="The IPv6 address of the target machine or device.  ↩
                              The IPv6 address can be found on Windows by running 'ipconfig' ↩
                               and looking for the 'IPv6 Address' field. For Linux run ' ↩
                              ifconfig' and look for 'inet6 addr' field. For OS X run ' ↩
                              ifconfig' and look for the 'inet6' field."/>

                <Range key="TargetPort"
                           value="12345"
                           min="0"
                           max="65535"
                           name="Target Port"
                           description="The target or destination port the network packet  ↩
                               is sent to."/>

                <Range key="Tag"
                           value="1"
                           min="0" max="4095"
                           name="VLAN identifier"
                           description="A 12-bit field specifying the VLAN to which the  ↩
                               frame belongs. The hexadecimal values of 0x000 and 0xFFF are  ↩
                               reserved. All other values may be used as VLAN identifiers,  ↩
                               allowing up to 4,094 VLANs. The reserved value 0x000  ↩
                               indicates that the frame does not belong to any VLAN; in this ↩
                                case, the 802.1Q tag specifies only a priority and is  ↩
                               referred to as a priority tag. On bridges, VLAN 1 (the  ↩
                               default VLAN ID) is often reserved for a management VLAN;  ↩
                               this is vendor-specific." />

                <String
                        key="LoggerPath"
                        value="logs/vlan/"
                        name="Logger Path"
                        description="The directory where Peach will save the log produced  ↩
                            when fuzzing." />

                <Strategy
                        key="Strategy"
                        value="Random"
                        name="Mutation Strategy"
                        description="The mutation strategy to use when fuzzing." />

                <String
                        key="PitLibraryPath"
                        value="."
                        name="Pit Library Path"
                        description="The path to the root of the pit library."/>
        </All>
</PitDefines>
```

# 37 Virtual Extensible Local Area Network (VXLAN)

- Peach Pit: VXLAN
- Direction: Client
- Supported Platforms: Linux

VXLAN (Virtual eXtensible Local Area Network) runs over the existing networking infrastructure and provides a means to "stretch" a Layer 2 network.

VXLAN is a Layer 2 overlay scheme over a Layer 3 network. Each overlay is termed a VXLAN segment.

Only VMs within the same VXLAN segment can communicate with each other.

Each VXLAN segment is scoped through a 24 bit segment ID [the VXLAN Network Identifier (VNI)]. This allows up to 16M VXLAN segments to coexist within the same administrative domain.

## 37.1 Specifications

| Specification | Title |
|---|---|
| http://tools.ietf.org/id/draft-mahalingam-dutt-dcops-vxlan-06.txt | VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks |

## 37.2 Use Cases

| Messages | Specification |
|---|---|
| VXLAN Frame | http://tools.ietf.org/id/draft-mahalingam-dutt-dcops-vxlan-06.txt |

## 37.3 Configuration

### 37.3.1 Target Configuration

Setting up VXLAN is a multi-step process. In order to simplify the setup, please refer to to VMware's official VLXAN deployment guide here.

The target must also have listeners configured for TCP and UDP reachable via IPv4 and IPv6.

### 37.3.2 Required Pit Configuration Changes

**TargetIPv6**
IPv6 address of the target host machine.

**SourceIPv6**
IPv6 address of the interface on the local machine.

**TargetIPv4**
IPv4 address of the target host machine.

**SourceIPv4**
IPv4 address of the interface on the local machine.

**SourceMAC**
MAC address on local machine.

**TargetMAC**
MAC address of target machine.

**OuterTargetIPv4**
IPv4 address of the target host machine.

**OuterSourceIPv4**
IPv4 address of the interface on the local machine.

**OuterSourceMAC**
MAC address on local machine.

**OuterTargetMAC**
MAC address of target machine.

**Tag**
VXLAN identifier specifying the VXLAN to which the frames belong.

**Interface**
Name of local interface (used for monitoring).

**SourcePort**
UDPv4 and/or TCPv4 port number of the local machine.

**TargetPort**
UDPv4 and/or TCPv4 port number of the target host machine.

#### 37.3.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 37.3.4 Configure Monitoring

Monitoring must be configured to provide fault detection, data collection, and automation as needed.

### 37.4 Running

#### 37.4.1 Single test debug run

```
peach -1 --debug VXLAN.xml
```

#### 37.4.2 Full test run

```
peach VXLAN.xml
```

### 37.5 Examples

**Example 37.1** Sample VXLAN Configuration File

Example configuration for sending vxlan packets.

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
  <All>
    <Hwaddr             key="OuterSourceMAC"
                        value="5254005335d3"
                        name="Source MAC Address"
                        description="Hardware address of the network interface on machine running Peach Fuzzer. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."/>

    <Ipv4               key="OuterSourceIPv4"
                        value="10.6.66.51"
                        name="Source IP Address"
                        description="The IPv4 address of the machine running Peach Fuzzer. The IPv4 address can be found on Windows by running 'ipconfig' and looking for the 'IPv4 Address' field. For Linux run 'ifconfig' and look for 'inet addr' field. For OS X run 'ifconfig' and look for the 'inet' field."/>

    <Hwaddr             key="OuterTargetMAC"
                        value="000000000000"
                        name="Target MAC Address"
                        description="Hardware address of the network interface on target machine or device. To find the hardware address on Windows, run 'ipconfig /all' and look for the ' Physical Address' field. For Linux run 'ifconfig' and look for the 'HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."  />

          <Ipv4         key="OuterTargetIPv4"
                        value="10.6.66.52"
                        name="Target IP Address"
                        description="The IPv4 address of the target machine or device. The IPv4 address can be found on Windows by running 'ipconfig' and looking for the 'IPv4 Address' field. For Linux run 'ifconfig' and look for 'inet addr' field. For OS X run 'ifconfig' and look for the 'inet' field." />

    <Hwaddr             key="SourceMAC"
                        value="6805ca0589fe"
                        name="Source MAC Address"
                        description="Hardware address of the network interface represented inside of the VXLAN encapsulation being sent from Peach Fuzzer. This address must not match OuterSourceMAC. To find the hardware address on Windows, run 'ipconfig /all' and look for the 'Physical Address' field. For Linux run 'ifconfig' and look for the ' HWaddr' field. For OS X run 'ifconfig' and look for the 'ether' field."/>

        <Ipv4           key="SourceIPv4"
                        value="127.0.0.1"
                        name="Source IPv4 Address"
```

```
                                description="The IPv4 address of the machine running Peach
                                    Fuzzer. The IPv4 address can be found on Windows by
                                    running 'ipconfig' and looking for the 'IPv4 Address'
                                    field. For Linux run 'ifconfig' and look for 'inet addr'
                                     field. For OS X run 'ifconfig' and look for the 'inet'
                                    field."/>

        <Ipv6           key="SourceIPv6"
                                value="::1"
                                name="Source IPv6 Address"
                                description="The IPv6 address of the machine running Peach
                                    Fuzzer. The IPv6 address can be found on Windows by
                                    running 'ipconfig' and looking for the 'IPv6 Address'
                                    field. For Linux run 'ifconfig' and look for 'inet6 addr
                                    ' field. For OS X run 'ifconfig' and look for the 'inet6
                                    ' field."/>

        <Range          key="SourcePort"
                                value="1234"
                                min="0" max="65535"
                                name="Source Port"
                                description="The source port the network packet originates
                                    from."/>

        <Hwaddr         key="TargetMAC"
                                value="2233ca00ff0f"
                                name="Target MAC Address"
                                description="Hardware address of the network interface
                                    represented inside of the VXLAN encapsulation being sent
                                     to a target. This address must not match OuterTargetMAC
                                    . To find the hardware address on Windows, run 'ipconfig
                                     /all' and look for the 'Physical Address' field. For
                                    Linux run 'ifconfig' and look for the 'HWaddr' field.
                                    For OS X run 'ifconfig' and look for the 'ether' field."
                                      />

    <Ipv4               key="TargetIPv4"
                                value="127.0.0.1"
                                name="Target IPv4 Address"
                                description="The IPv4 address of the target machine or
                                    device. The IPv4 address can be found on Windows by
                                    running 'ipconfig' and looking for the 'IPv4 Address'
                                    field. For Linux run 'ifconfig' and look for 'inet addr'
                                     field. For OS X run 'ifconfig' and look for the 'inet'
                                    field." />

        <Ipv6           key="TargetIPv6"
                                value="::1"
                                name="Target IPv6 Address"
                                description="The IPv6 address of the target machine or
                                    device. The IPv6 address can be found on Windows by
                                    running 'ipconfig' and looking for the 'IPv6 Address'
                                    field. For Linux run 'ifconfig' and look for 'inet6 addr
                                    ' field. For OS X run 'ifconfig' and look for the 'inet6
                                    ' field."/>

    <Range              key="TargetPort"
                                value="1234"
                                min="0" max="65535"
                                name="Target Port"
                                description="The target or destination port the network
                                    packet is sent to."/>
```

```
      <Range          key="Vni"
                      value="42"
                      min="0" max="16777215"
                      name="VXLAN Network Identifier"
                      description="VXLAN Segment ID/VXLAN Network (VNI) - this is ↩
                          a 24 bit value used to designate the individual VXLAN  ↩
                          overlay network on which the communicating VMs are  ↩
                          situated.  VMs in different VXLAN overlay networks  ↩
                          cannot communicate with each other."/>

      <Range          key="Tag"
                      value="2"
                      min="0"
                      max="4095"
                      name="VLAN identifier"
                      description="A 12-bit field specifying the VLAN to which  ↩
                          the frame belongs. The hexadecimal values of 0x000 and 0 ↩
                          xFFF are reserved. All other values may be used as VLAN  ↩
                          identifiers, allowing up to 4,094 VLANs. The reserved  ↩
                          value 0x000 indicates that the frame does not belong to  ↩
                          any VLAN; in this case, the 802.1Q tag specifies only a  ↩
                          priority and is referred to as a priority tag. On  ↩
                          bridges, VLAN 1 (the default VLAN ID) is often reserved  ↩
                          for a management VLAN; this is vendor-specific." />

      <String         key="LoggerPath"
                      value="logs/vxlan/"
                      name="Logger Path"
                      description="The directory where Peach will save the log  ↩
                          produced when fuzzing." />

      <Strategy       key="Strategy"
                      value="Random"
                      name="Mutation Strategy"
                      description="The mutation strategy to use when fuzzing." />

      <String         key="PitLibraryPath"
                      value="."
                      name="Pit Library Path"
                      description="The path to the root of the pit library."/>
</All>

<Linux>
  <Iface              key="Interface"
                      value="eth0"
                      name="Network Interface"
                      description="The network interface to transmit packets over ↩
                          . For Windows, the network interfaces can be shown by  ↩
                          running 'ipconfig'. On Linux and OS X, the network  ↩
                          interfaces can be shown by running the command 'ifconfig ↩
                          '."/>
</Linux>

<OSX>
   <!-- This can't run on OS X because it uses the RawEther publisher. -->
</OSX>

<Windows>
  <!-- This can't run on windows because it uses the RawEther publisher. -->
</Windows>
```

```
</PitDefines>
```

# 38 Wifi (802.11)

- Peach Pit: Wifi
- Direction: Client
- Supported Platforms: Linux (Ubuntu Server 14.04 LTS)
- Supported Wireless Chipsets: RaLink Chipset

IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6, 5 and 60 GHz frequency bands.

## 38.1 Specifications

| Specification | Title |
|---|---|
| IEEE 802.11-2012 | IEEE 802.11-2012 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) |

## 38.2 Use Cases

| Messages | Specification |
|---|---|
| Probe Response | IEEE 802.11-2012 |
| Authentication | IEEE 802.11-2012 |
| Association Response | IEEE 802.11-2012 |
| Reassociation Response | IEEE 802.11-2012 |
| Deauthentication | IEEE 802.11-2012 |
| Disassociation | IEEE 802.11-2012 |
| Data Frame | IEEE 802.11-2012 |
| Acknowledgement (ACK) | IEEE 802.11-2012 |
| Ready-To-Send (RTS) | IEEE 802.11-2012 |
| Clear-To-Send (CTS) | IEEE 802.11-2012 |
| Action Frame | IEEE 802.11-2012 |

| Supported Security Modes | Specification |
|---|---|
| Plain (No Security) | IEEE 802.11-2012 |

## 38.3 Supported Wireless Adapters

The Wifi Pit supports USB Wireless devices based on the RaLink chipsets. At the time of writing the following devices are known to use the RaLink chipsets and work with the Wifi Pits:

- Panda Wireless PAU03
- Protronix 802.11N/G Wireless USB Adapter
  - Supports external antenna

Both wireless adapters are available from Amazon and other retailers.

**Chipsets Known Not to Work:**

All other chipsets tested failed to work. This included Atheros and Realtek chipsets. The most common issues identified were related to wireless packet injection.

## 38.4 Supported Operating Systems

Deja vu Security recommends using Ubuntu Server 14.04 LTS on a physical machine, NOT a virtual machine, to perform Wifi fuzzing. During testing stability issues were identified when running with older Linux kernels or in a virtualized environment.

## 38.5 Tested Wireless Stacks

During testing of the Wifi Pit the following devices/stacks were tested for compatibility:

- Windows 8.1
- Linux (Ubuntu Server 14.04 LTS)
- Android (Multiple versions)
- iPhone 5S
- iPhone 5

It was found during testing that occasional timing issues would prevent association from working on the first try. This was most prevalent with Windows 8.1. In the case that the first iteration does not complete, re-running the test typically worked. If association is still failing after five attempts, contact Peach support for assistance.

## 38.6 Configuration

### 38.6.1 Target Configuration

A method for triggering the target to start the Wifi association is needed. The Wifi Pit must then be configured to trigger the association.

### 38.6.2 Required Pit Configuration Changes

**Interface**
Name of local wireless interface

**TargetMAC**
MAC address of target machine

**SourceMAC**
MAC address on local machine

**Ssid**
SSID for Wireless network

**Channel**
Channel number between 1 and 14. ChannelFrequency must match the selected channel.

**ChannelFrequency**
Channel frequency. Must match the Channel parameter using the following table.

| Channel | Frequency |
|---|---|
| 1 | 2412 |
| 2 | 2417 |
| 3 | 2422 |
| 4 | 2427 |
| 5 | 2432 |
| 6 | 2437 |

| Channel | Frequency |
|---|---|
| 7 | 2442 |
| 8 | 2447 |
| 9 | 2452 |
| 10 | 2457 |
| 11 | 2462 |
| 12 | 2467 |
| 13 | 2472 |
| 14 | 2484 |

#### 38.6.3 Optional Pit Configuration Changes

**Strategy**
Fuzzing strategy Peach will use for testing.

**LoggerPath**
Path to folder where logs will be stored.

**Path**
Path to the relative base directory where all pits are located.

#### 38.6.4 Configure Additional Monitors

Add monitors to agent as needed

### 38.7 Running

#### 38.7.1 Single test debug run

**Fuzzing Wifi Client Association**

```
peach -1 --debug Wifi_Client.xml
```

#### 38.7.2 Full test run

**Fuzzing Wifi Client Association**

```
peach Wifi_Client.xml
```

### 38.8 Examples

**Example 38.1** Wifi Client Association Fuzzing, Linux

This example will show a working configuration using two Ubuntu Server 14.04 LTS machines. One will host our fuzzer and will be referred to as the "fuzzer" or "fuzzing" machine. The second machine is the "target" machine.

**Fuzzing Machine Configuration**

The fuzzing machine is configured with the latest version of Peach Professional and a supported wireless device. It must also have IP connectivity to the target machine.

*Install wireless-tools package:*

```
sudo apt-get install wireless-tools
```

**Target Machine Configuration**
The target machine is configured with the latest version of Peach Professional, the wireless tools package, and a Linux supported wireless device.
*Install wireless-tools package:*

```
sudo apt-get install wireless-tools
```

*Configure Wireless Interface:*
A wireless network interface must be configured in `/etc/network/interfaces`. The following is an example configuration that assumes the wireless device is `wlan0`. This configuration should be appended to the `/etc/network/interfaces` file.

```
iface wlan0 inet dhcp
        wireless-essid PeachWifi
```

**Configure Wifi Pit**
*Wifi_Client.xml*
To configure the Wifi Pit we will need to add a remote agent that will trigger our Ubuntu Server target to start a Wifi association. This is done using the `ifup` command. We will also need to stop the association at the end of the iteration. For Ubuntu Server 14.04 LTS this is done by killing the DHCP client process (dhclient) and then calling the `ifdown` command.
The interface name we will allow to be configurable by adding a *TargetInterface* configuration value. An entry in the configuration file will be needed for it.

```
  <Agent name="TargetAgent" location="tcp://10.0.1.71:9001">
    <Monitor class="RunCommand">
      <Param name="Command" value="/usr/bin/killall" />
      <Param name="Arguments" value="dhclient" />
      <Param name="StartOnCall" value="Cleanup" />
    </Monitor>

    <Monitor class="RunCommand">
      <Param name="Command" value="/sbin/ifdown" />
      <Param name="Arguments" value="##TargetInterface##" />
      <Param name="StartOnCall" value="Cleanup" />
    </Monitor>

    <!-- Connect -->

    <Monitor class="Process">
      <Param name="Executable" value="/sbin/ifup" />
      <Param name="Arguments" value="##TargetInterface##" />
      <Param name="StartOnCall" value="Connect" />
      <Param name="NoCpuKill" value="true" />
      <Param name="WaitForExitTimeout" value="0" />
    </Monitor>
  </Agent>
```

The agent must be added to the *Test* element using this line:

```
    <Agent ref="TargetAgent" />
```

Here is the completed *Wifi_Client.xml* file:

```
<?xml version="1.0" encoding="utf-8"?>
<Peach xmlns="http://peachfuzzer.com/2012/Peach" xmlns:xsi="http://www.w3.org/2001/ ↩
    XMLSchema-instance"
    xsi:schemaLocation="http://peachfuzzer.com/2012/Peach peach.xsd">

  <Include ns="Wifi" src="file:../_Common/Models/Net/Wifi_State.xml"/>

  <Agent name="Local">

    <!-- Startup -->
```

```
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifconfig" />
    <Param name="Arguments" value="##Interface## down" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/iwconfig" />
    <Param name="Arguments" value="##Interface## mode ad-hoc" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifconfig" />
    <Param name="Arguments" value="##Interface## up" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifconfig" />
    <Param name="Arguments" value="##Interface## down" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/iwconfig" />
    <Param name="Arguments" value="##Interface## mode monitor" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifconfig" />
    <Param name="Arguments" value="##Interface## up" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>
  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/iwconfig" />
    <Param name="Arguments" value="##Interface## chan ##Channel##" />
    <Param name="StartOnCall" value="Init" />
  </Monitor>

  <!-- Shutdown -->

  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifconfig" />
    <Param name="Arguments" value="##Interface## down" />
    <Param name="When" value="OnEnd" />
  </Monitor>
</Agent>

<Agent name="TargetAgent" location="tcp://##TargetAgentIp##:9001">
  <Monitor class="RunCommand">
    <Param name="Command" value="/usr/bin/killall" />
    <Param name="Arguments" value="dhclient" />
    <Param name="StartOnCall" value="Cleanup" />
  </Monitor>

  <Monitor class="RunCommand">
    <Param name="Command" value="/sbin/ifdown" />
    <Param name="Arguments" value="##TargetInterface##" />
    <Param name="StartOnCall" value="Cleanup" />
  </Monitor>

  <!-- Connect -->

  <Monitor class="Process">
```

```
      <Param name="Executable" value="/sbin/ifup" />
      <Param name="Arguments" value="##TargetInterface##" />
      <Param name="StartOnCall" value="Connect" />
      <Param name="NoCpuKill" value="true" />
      <Param name="WaitForExitTimeout" value="0" />
    </Monitor>
  </Agent>

  <Test name="Default">
    <Exclude xpath="//Radiotap"/>
    <Exclude xpath="//ReceiverAddress"/>

    <StateModel ref="Wifi:AP"/>

    <Agent ref="Local" />
    <Agent ref="TargetAgent" />

    <Publisher class="Wifi">
        <Param name="Interface" value="##Interface##" />
      <Param name="TargetMac" value="##TargetMAC##"/>
      <Param name="SourceMac" value="##SourceMAC##"/>
      <Param name="Timeout" value="10000"/>
      <Param name="ApAuthTimeout" value="10000" />
    </Publisher>

        <Strategy class="##Strategy##" />

    <Logger class="File">
      <Param name="Path" value="##LoggerPath##"/>
    </Logger>
  </Test>
</Peach>
```

*Wifi_Client.xml.config*
For the Wifi configuration file, the only parameters that must be updated are:

**TargetInterface**
The targets wireless interface name. This configuration value was added with our Agent configuration performed during this example. This configuration option does not ship with the Wifi pit.

**Interface**
Local wireless interface. This must be a supported wireless device.

**TargetMAC**
The hardware address of the targets wireless interface. Peach will only respond to this MAC address.

**SourceMAC**
The hardware address of the local wireless interface.

Here is an example, completed configuration file:

```
<?xml version="1.0" encoding="utf-8"?>
<PitDefines>
  <All>
    <Ipv4 key="TargetAgentIp"
          value="10.0.1.71" />

    <Iface key="TargetInterface" value="wlan0" />

    <Iface key="Interface"
           value="wlan0"
           name="Local Wireless Interface"
           description="The local wireless interface name."/>
```

```
<Hwaddr key="TargetMAC"
        value="7cdd9047b053"
        name="Target MAC Address"
        description="MAC address of target wireless device."/>

<Hwaddr key="SourceMAC"
        value="7cdd906146f5"
        name="Source MAC Address"
        description="MAC address of local wireless device."/>

<!--

Channel Frequency Map

C   Freq    NA  JP  World
1   2412    Yes     Yes     YesD
2   2417    Yes     Yes     YesD
3   2422    Yes     Yes     YesD
4   2427    Yes     Yes     YesD
5   2432    Yes     Yes     Yes
6   2437    Yes     Yes     Yes
7   2442    Yes     Yes     Yes
8   2447    Yes     Yes     Yes
9   2452    Yes     Yes     Yes
10  2457    Yes     Yes     Yes
11  2462    Yes     Yes     Yes
12  2467    No      Yes     Yes
13  2472    No      Yes     Yes
14  2484    No  11b No

-->

<Range key="Channel"
       value="2"
       min="1" max="14"
       name="Wireless Channel"
       description="The wireless channel to broadcast on."/>

<Range key="ChannelFrequency"
      value="2417"
       min="2412" max="2484"
       name="Wireless Channel Frequency"
       description="The wireless channel frequency to broadcast on. Must match channel
           number. Map: 1 - 2412; 2 - 2417; 3 - 2422; 4 - 2427; 5 - 2432; 6 - 2437; 7 -
           2442; 8 - 2447; 9 - 2452; 10 - 2457; 11 - 2462; 12 - 2467; 13 - 2472; 14 -
           2484." />

<String key="Ssid"
        value="PeachWifi"
        name="Wireless SSID"
        description="The wireless station identifier."/>

<String key="LoggerPath"
        value="logs/wifi_client/"
        name="Logger Path"
        description="The directory where Peach will save the log produced when fuzzing.
            " />

<Strategy key="Strategy"
          value="Random"
          name="Mutation Strategy"
```

```
              description="The mutation strategy to use when fuzzing." />

    <String key="PitLibraryPath"
            value="."
            name="Pit Library Path"
            description="The path to the root of the pit library."/>

    <!-- Do not modify values below this line -->

    <Ipv4 key="TargetIPv4" value="255.255.255.255" name="Target IP" description="Advanced option, do not modify."/>
    <Range key="TargetPort" value="68" min="0" max="65535" name="Target Port" description="Advanced option, do not modify."/>

    <Ipv4 key="SourceIPv4" value="192.168.1.1" name="Source IP" description="Advanced option, do not modify."/>
    <Range key="SourcePort" value="67" min="0" max="65535" name="Source Port" description="Advanced option, do not modify."/>

    <Ipv4 key="AssignedIPv4" value="192.168.1.55" name="Assigned IP" description="Advanced option, do not modify."/>
    <Ipv4 key="RouterIP" value="192.168.1.1" name="Router IP" description="Advanced option, do not modify."/>
    <Ipv4 key="BroadcastIP" value="192.168.1.255" name="Broadcast IP" description="Advanced option, do not modify."/>
    <Ipv4 key="DHCPServerIP" value="192.168.1.1" name="DHCP Server IP" description="Advanced option, do not modify."/>
    <Ipv4 key="SubnetMask" value="255.255.255.0" name="Subnet Mask" description="Advanced option, do not modify."/>
    <String key="DomainName" value="localdomain" name="Domain Name" description="Advanced option, do not modify."/>
    <Ipv4 key="DNS" value="192.168.1.2" name="DNS" description="Advanced option, do not modify."/>
    <Ipv4 key="NetBios" value="192.168.1.2" name="NetBios" description="Advanced option, do not modify."/>

  </All>
</PitDefines>
```