Sulley fuzzer learning---5

行之 / 2018-08-22 09:45:43 /

前言

Sulley fuzzer的应用2.
原文链接：https://resources.infosecinstitute.com/fuzzing-vulnserver-with-sulley-part-3/#gref

Fuzzing Vulnserver with Sulley

介绍

Vulnserver是一个易受攻击的服务器，由Stephen
Bradshaw撰写，他的博客位于此处：grey-corner。这个服务器是故意编写的，因此我们可以学习fuzz测试。如果我们没有任何现有的漏洞来测试它，那么很难学会fuzz测试

确定目标
我们已经这样做了，因为我们即将fuzz Vulnserver。因此，目标已经被识别，并且它是在端口9999上运行的Vulnserver。

识别输入
在执行任何其他操作之前，我们必须识别应用程序接受和处理的所有输入。我们知道存在漏洞，因为应用程序采用格式错误的输入并在不进行清理的情况下对其进行处理

```
# telnet localhost 9999
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

我们可以看到服务器确实正在侦听端口9999，一旦我们连接到它，我们就执行了HELP命令，该命令显示了服务器支持的所有有效命令。

生成fuzz数据
在我们识别出所有输入向量之后，我们必须生成要发送到目标应用程序的无效输入数据（在我们的例子中，是Vulnserver）。

执行fuzz数据
当我们生成fuzz数据后，是时候将它发送到端口9999上的Vulnserver了。

检测异常情况
我们还必须监视Vulnserver的崩溃。当服务器崩溃时，我们必须保存导致崩溃的输入数据，以便以后进行分析。

确定可利用性
在fuzz测试之后，要通过找到的崩溃并加以利用，我们必须手动检查导致Vulnserver崩溃的所有已保存数据。

启动fuzz测试

在这一步中，我们将解释一个文件vulnserver.py，它将用作Sulley
fuzzer的输入，并将实际执行所有工作。要开始对Vulnserver进行fuzz测试，我们需要在客户操作系统上启动network_monitor.py和process_monitor.py（实际让Vulnser
fuzer应该已经安装并且功能正常）：

```
<a href="file:///C:/">C:\</a>> cd C:\sulley\
C:\sulley> mkdir audits\vulnserver\
```

```
C:\sulley> python network_monitor.py -d 0 -f "src or dst port 9999" -P audits\vulnserver\
C:\sulley> python process_monitor.py -c audits\vulnserver.crashbin -p vulnserver.exe
```

我们创建了一个新目录，用于存储PCAP文件 -
从Vulnserver传输的每个数据包都将记录在PCAP文件中。每个PCAP文件将包含来自fuzz过程的一次迭代的信息，因此将有相当多的PCAP文件。但不要担心，我们只会提取

在fuzzer操作系统上，我们需要使用以下命令启动fuzz测试过程，其中vulnserver.py是Sulley fuzzer测试框架中的输入文件，描述了fuzz测试过程的所有方面：

```
# export PYTHONPATH="$PYTHONPATH:/home/user/sulley/
# python requests/vulnserver.py
```

我们已经将Sulley安装目录添加到PYTHONPATH中，因此python可以找到它需要正常运行的所有库。之后我们开始真正的fuzz测试过程。

在启动fuzz测试过程后，我们可以在URL地址http://127.0.0.1:26000
上的Web界面中观察fuzz测试进度。这是一个Sulley内部Web服务器，向我们显示fuzz过程完整性和导致崩溃的输入文件。

我们可以看到很多崩溃; 左边的数字表示fuzz过程的迭代（因此也是写入audits \ vulnserver
\目录的PCAP文件的名称，我们可以在其中找到使易受攻击的服务器崩溃的数据包）。我们可以看到有很多崩溃，其中EIP被输入数据覆盖，如0x41414141或0x20202020，
到目前为止，我们已经看到了我们应该得到的结果，但实际上在所有fuzz的过程中vulnerver.py做了什么？

vulnserver.py输入文件

首先让我们展示整个vulnserver.py文件：

```
#!/usr/bin/python
from sulley import *
import sys
import time
""" Receive banner when connecting to server. """
def banner(sock):
sock.recv(1024)
""" Define data model. """
s_initialize("VulnserverDATA")
s_group("commands", values=['HELP', 'STATS', 'RTIME', 'LTIME', 'SRUN', 'TRUN', 'GMON', 'GDOG', 'KSTET', 'GTER', 'HTER', 'LTER'
s_block_start("CommandBlock", group="commands")
s_delim(' ')
s_string('fuzz')
s_static('\r\n')
s_block_end()
""" Keep session information if we want to resume at a later point. """
s = sessions.session(session_filename="audits/vulnserver.session")
""" Define state model. """
s.connect(s_get("VulnserverDATA"))
""" Define the target to fuzz. """
target = sessions.target("192.168.1.126", 9999)
target.netmon = pedrpc.client("192.168.1.126", 26001)
target.procmon = pedrpc.client("192.168.1.126", 26002)
target.procmon_options = {
"proc_name" : "vulnserver.exe",
"stop_commands" : ['wmic process where (name="vulnserver.exe") delete'],
"start_commands" : ['C:\\Users\\eleanor\\Desktop\\vulnserver\\vulnserver.exe'],
}
""" grab the banner from the server """
s.pre_send = banner
""" start fuzzing - define target and data """
s.add_target(target)
s.fuzz()
```

首先，我们定义一个函数，在与漏洞服务器的每个连接上接收banner。这是必需的，因为在启动与易受攻击服务器的连接时，服务器向我们发送欢迎消息，在尝试将任何数

使用s_group，我们不必单独定义每个命令;
我们只定义命令的名称，然后是空格分隔符和参数，它将被fuzz。我们已经描述了数据模型的声明，它实际上并没有做任何事情。我们必须提供一个数据模型，让Sulley知道

接下来是一个状态模型，它描述了Sulley在fuzz测试时必须经历的状态。在我们的简单示例中，我们不需要很多状态，只需要一个实际使用先前定义的数据模型的状态，并按
以下是target声明。让我们再看一下这段代码：

```
target = sessions.target("192.168.1.126", 9999)
target.netmon = pedrpc.client("192.168.1.126", 26001)
target.procmon = pedrpc.client("192.168.1.126", 26002)
target.procmon_options = {
"proc_name" : "vulnserver.exe",
```

```
"stop_commands" : ['wmic process where (name="vulnserver.exe") delete'],
"start_commands" : ['C:\\Users\\eleanor\\Desktop\\vulnserver\\vulnserver.exe'],
}
```

这些行声明目标被fuzz。我们已经说过，在目标机器上我们应该运行网络和处理器监视器 -
这两个将在端口26001和26002上侦听。所以这里我们声明这两个监视器可以访问哪些IP和端口。sessions.target参数指定发送fuzz数据的目标，procmon_options指定可用

在vulnserver.py结束时，我们只是抓住了Vulnserver的banner，现在我们可以开始fuzz测试了。

```
""" start fuzzing - define target and data """
s.add_target(target)
s.fuzz()
```

结果

Sulley能够在命令KSTET，TRUN，GMON，GTER，HTER和LTER中找到所有六个漏洞。它会多次检测到一些漏洞。让我们来看看Sulley在fuzz漏洞服务器时发现的漏洞数量

```
TRUN : 3x
GMON : 1x
KSTET: 4x
GTER : 2x
HTER : 1x
LTER : 1x
```

Sulley发现了3个输入案例，它可以在fuzz TRUN命令时使漏洞服务器崩溃，1个输入案例在fuzz
GMON命令时崩溃。输入测试用例在fuzz命令LTER时漏洞服务器崩溃情况如下所示：

```
LTER /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\r\n
```

结论

我们知道如何使用Sulleyfuzz易受攻击的服务器。我们还看到Sulley成功找到了漏洞服务器中存在的所有六个漏洞。Sulley是一个很好的工具，用于fuzz某些软件的新漏洞。

上一篇：【2018年 网鼎杯CTF 第一场... 下一篇：blackhat议题深入 | ph...