

HTTP 应用服务业务测试 配置手册

北京网测科技

2021/11/12

目录

1. 文档说明.....	1
2. 配置事务列表并运行事务用例.....	2
2.1 创建事务列表.....	2
2.1.1 录制事务访问过程.....	3
2.1.1.1 HAR 文件方式录制事务.....	4
2.1.1.2 Pcap 文件方式录制事务.....	7
2.1.2 创建 HTTP 事务列表.....	12
2.1.3 完善 HTTP 事务列表.....	14
2.2 创建事务用例.....	24
2.3 运行事务用例.....	26
2.4 抓包查看报文.....	28
3. 事务列表的其他功能.....	31
3.1 业务组功能.....	32
3.5.1 创建 HTTP 事务列表.....	32
3.5.2 创建 HTTP 事务用例并运行.....	33
3.5.3 抓包查看报文.....	34

1. 文档说明

Supernova 系列测试仪，支持测试任何以 HTTP 协议为基础，开发部署的 Web 应用和网站，既可以测试其性能，也可以模拟其业务处理的过程，测试业务逻辑。测试对象如下：

1. 电子政务网站，如政府官网、政务之窗、12306 购票系统等。
2. 电子商务网站，如淘宝、京东、苏宁、网票网等。
3. 银行网银系统，如各大银行的网银系统，信用卡系统等。
4. 各类企业官网，如顺丰、邮政、中石油、中石化、移动、电信、联通等
5. 各种 Web 应用，如 Mantis、Jenkins、客户管理系统、进销存系统等。
6. 基于 HTTP 的手机 APP，现在大多数手机应用都是依靠 HTTP 协议与服务器通信，所以也可以模拟手机 APP，测试与服务器的通信。

单独测试 HTTP 的性能，HTTP 新建、并发、吞吐、请求用例就可以满足要求。如果要测试业务逻辑，比如登录、下单购买、取消订单、退出登录，就要配置专门的 HTTP 事务用例。

HTTP 事务支持网关设备测试和应用服务器测试，本文档主要介绍应用服务器测试下配置 HTTP 事务（即 HTTP 应用服务业务测试）的方法和过程。HTTP 应用服务业务测试的配置涉及到 HTTP 协议的专业知识、Web 应用的研发逻辑、正则表达式的应用，而且随着 Web 服务器的版本升级和接口变化，需要不断对配置用例进行修改和升级，所以有任何问题，请联系我们的售前或售后支持人员。

2. 配置事务列表并运行事务用例

我们要运行 HTTP 应用服务业务测试用例，必须先要在系统的“对象”→“Web 测试项目”，创建一个 Web 测试项目，并在右侧窗口中创建一个 Web 功能模块来配置 HTTP 事务列表，创建 HTTP 应用服务业务测试用例时在配置中引用新创建的事务列表。

2.1 创建事务列表

配置一个 HTTP 事务列表，需要以下几个步骤：

(1) 录制事务访问过程

测试仪支持导入事务 Pcap 文件和 HAR 文件，可以采用两种方式录制事务访问过程：

- a) 使用 Firefox 或 Chrome 浏览器访问页面，录制事务流程。打开 Firefox 或者 Chrome 之后，按 F12 启用调试界面，进入网络选项卡，启用持续日志，禁用缓存，所有页面访问都完成后，右键单击流量列表，保存流量为 HAR 文件；
- b) 使用浏览器或者手机 APP，在访问 Web 应用或者网站的同时，用软件工具 Wireshark 或者 Tcpdump 进行报文过滤和捕获，把 HTTP 业务访问过程保存为 pcap 文件。

(2) 创建 HTTP 事务列表

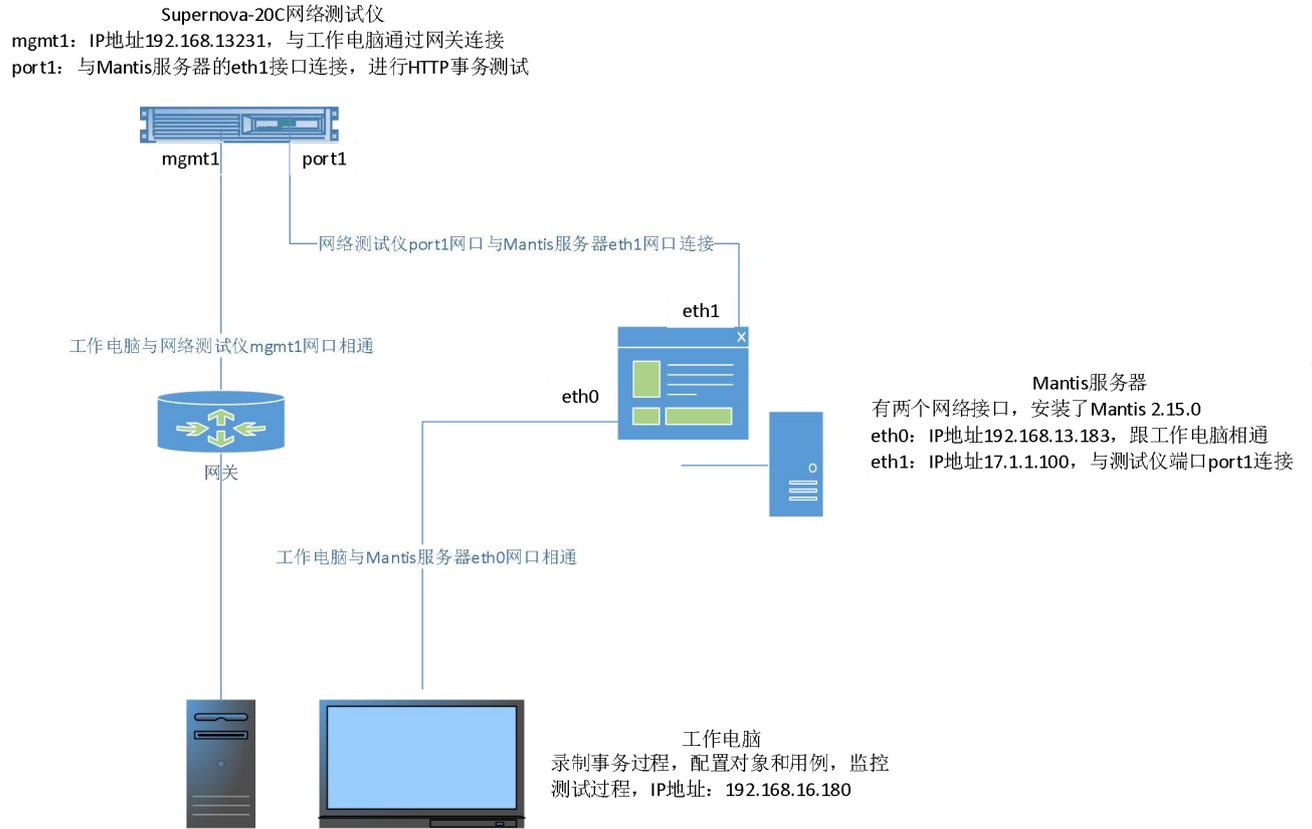
在 Supernova 测试仪的 Web 界面，创建一个 Web 功能模块，把保存的 HAR 文件或者 Pcap 文件上传导入，生成一个 HTTP 事务列表的初始配置。

(3) 完善 HTTP 事务列表

初始的 HTTP 事务配置，只是生成了一个框架，并不能直接运行，要对其修改完善，才能模拟出更多并发的业务逻辑。比如登录、下单购买、取消订单、退出登录的业务逻辑，如果不做修改，每个虚拟用户将提交一个订单，但取消的订单号，一直是原始的录制事务访问时的订单号，后面提交的订单，都不会取消掉。

2.1.1 录制事务访问过程

为了便于理解整个配置和测试过程，网络拓扑如下所示：

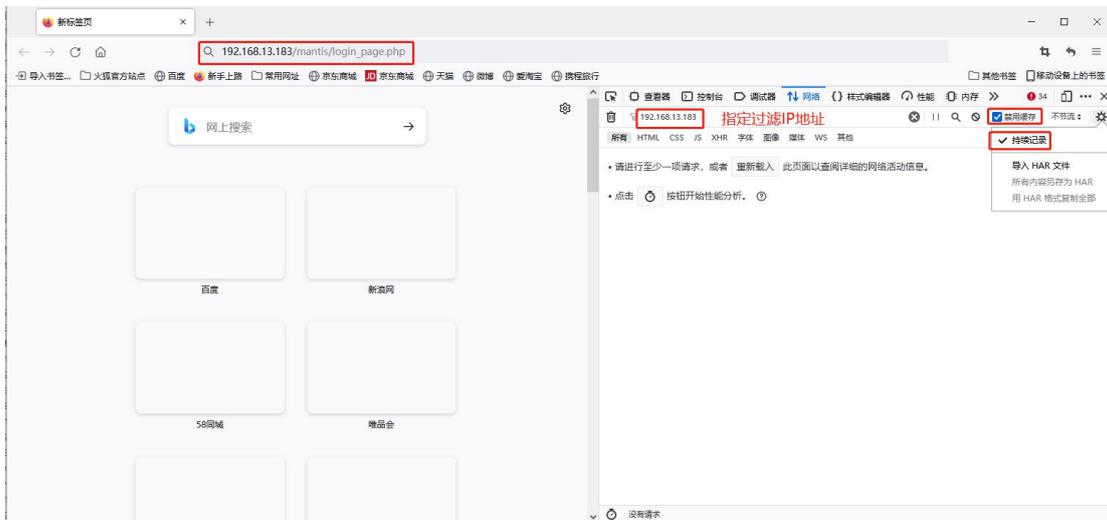


2.1.1.1 HAR 文件方式录制事务

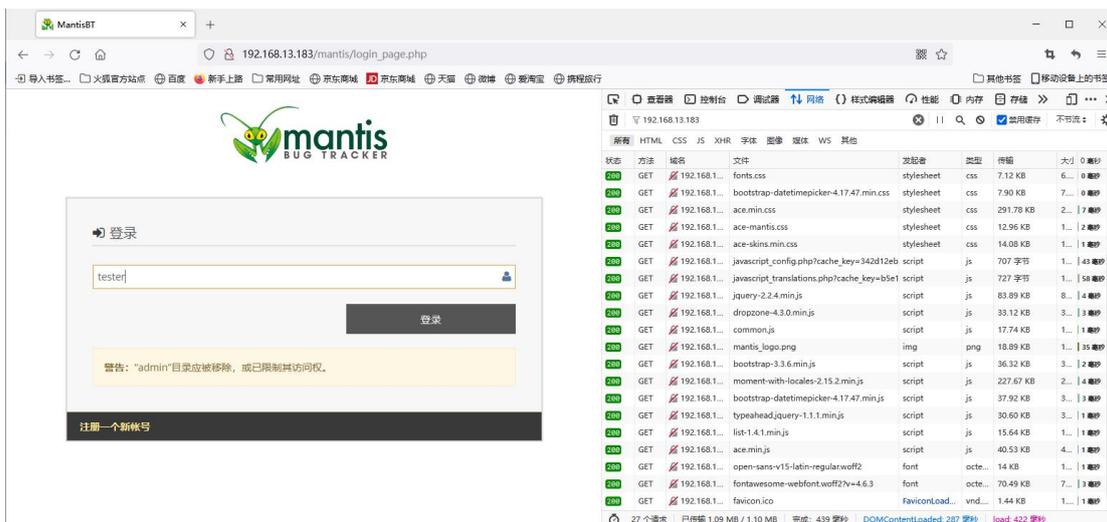
打开 Firefox 之后，按 F12 启用调试界面，进入网络选项卡，启用持续日志，禁用缓存，所有页面访问都完成后，右键单击流量列表，保存流量为 HAR 文件。

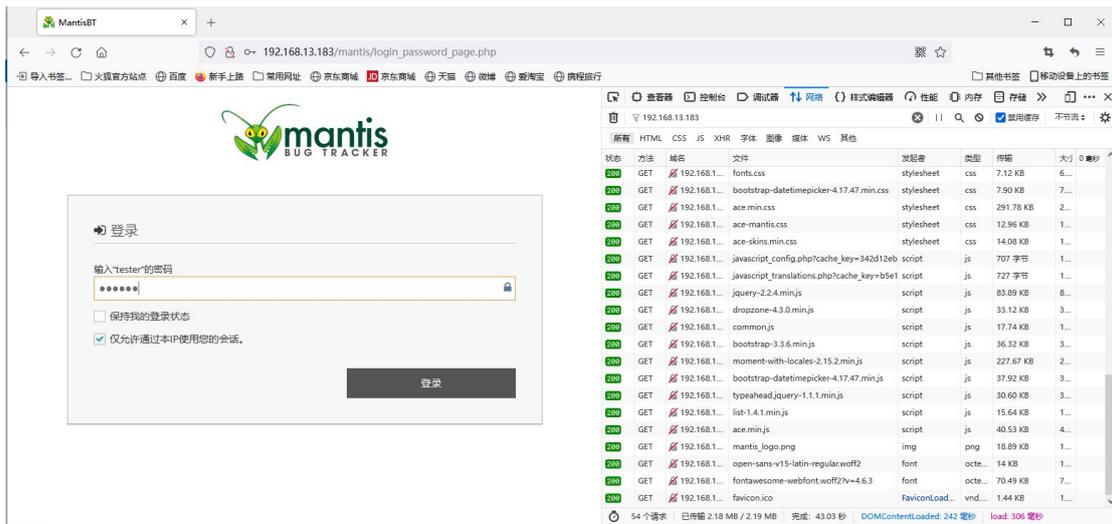
我们以录制一个 mantis 事务为例，介绍具体过程。Mantis 是研发人员常用的 Bug 追踪系统，我们在 Mantis 上创建了测试人员账号 tester/tester，具体的业务流程是：用 tester/netitest 账号登录 Mantis 系统，创建一个 Bug，然后再删除它，最后退出。

- 1) 打开 Firefox 浏览器输入 mantis 服务器网址 192.168.13.183/mantis/login_page.php，按 F12 启用调试界面，进入网络选项卡，输入 ip 地址过滤并启用持续记录，禁用缓存。

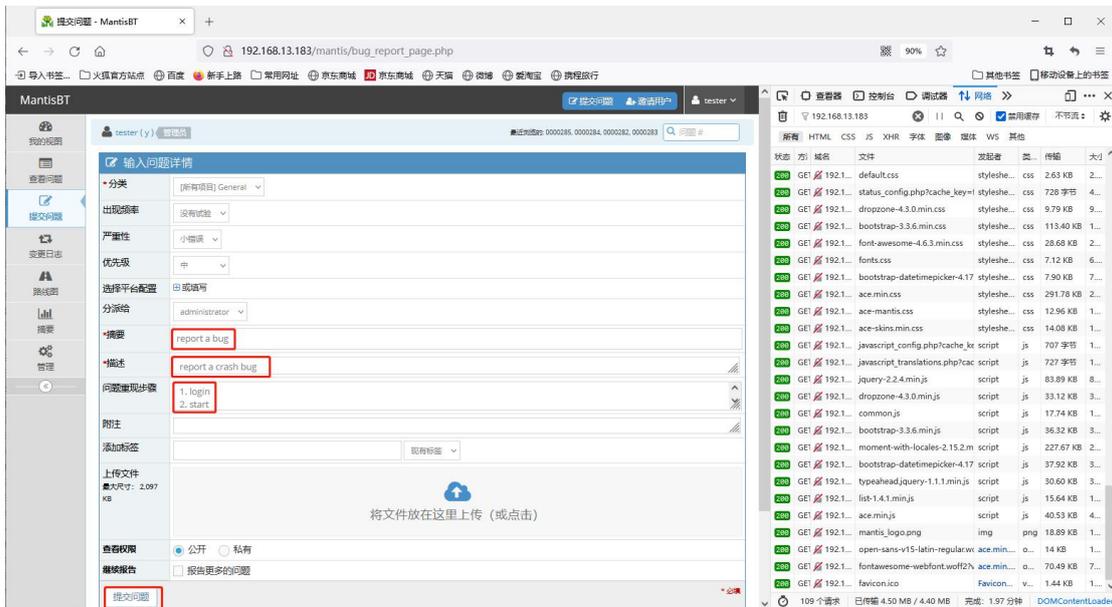


- 2) 用账号密码 tester/tester 登录 mantis，并进行新增 bug，删除 bug，退出登录操作。

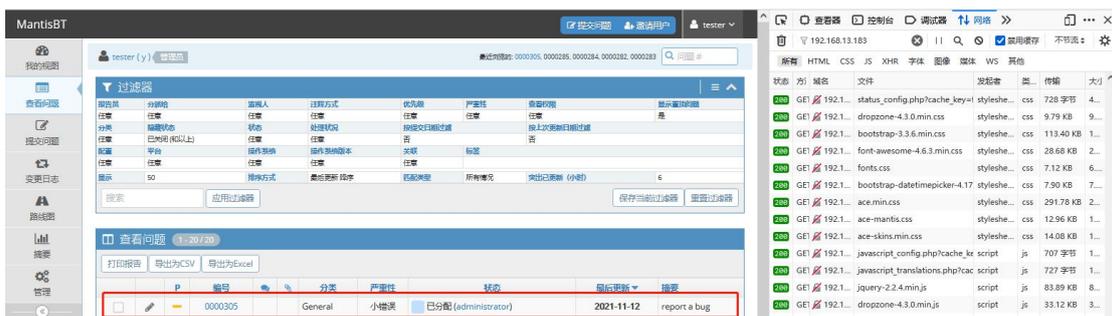




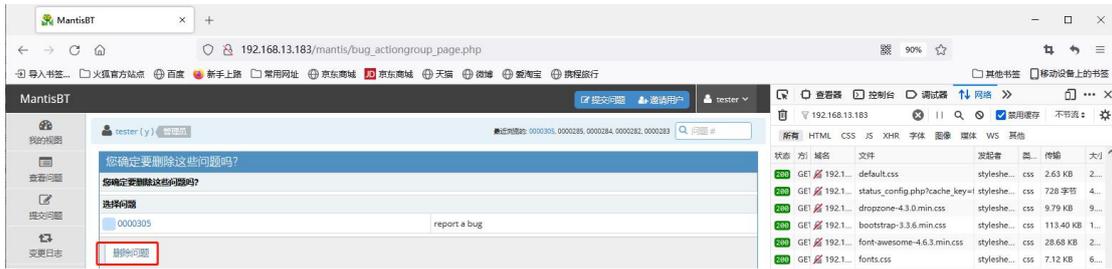
创建并提交一个新的 bug，摘要填写“report a bug”，描述填写“report a crash bug”，问题重现步骤填写“1. login 2. start”，点击提交问题。



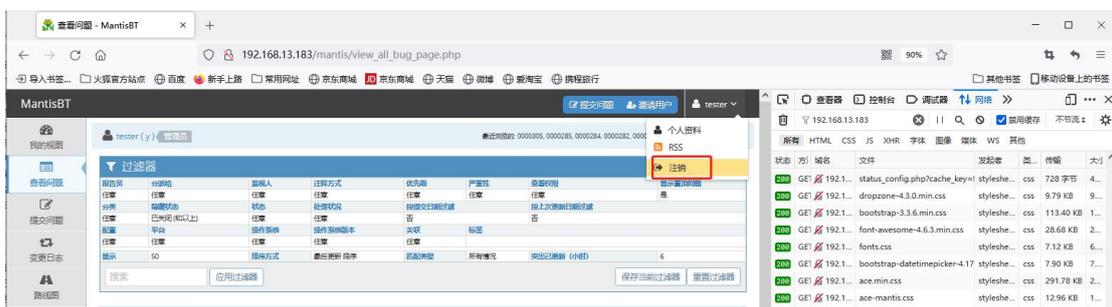
点击提交问题后，页面就会提示提交成功，会很快跳转到显示页面。几秒后，Mantis 自动跳转到显示页面，显示刚刚提交的 Bug。



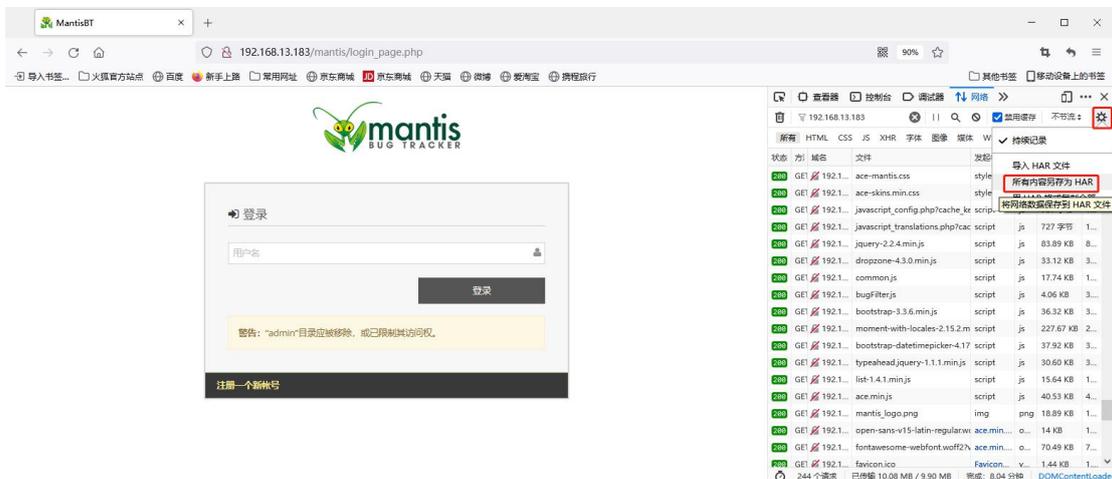
选中刚才提交的 Bug，选中“删除”选项，点击确定，开始删除这个 Bug。点击删除问题，Bug 将被删除，Mantis 自动跳转到 Bug 显示界面。



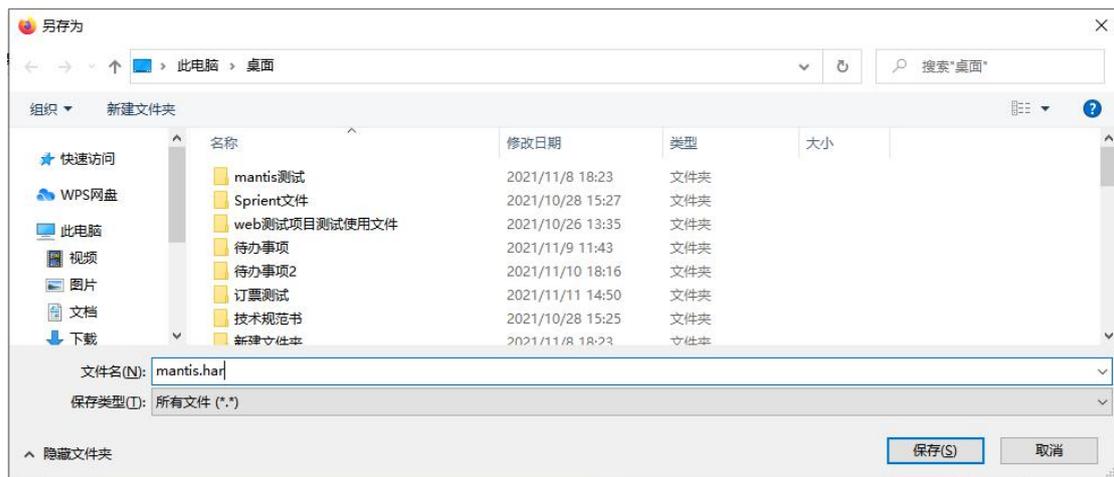
点击右上角的 tester 账号的下拉菜单，点击注销，退出系统。一个完整的 HTTP 事务流程就完成了，包括 Mantis 登录，提交 Bug，删除 Bug，退出登录。



3) 所有页面访问都完成后，右键单击流量列表或点击设置按钮，保存所有内容为 HAR 文件。



输入文件名，选择保存目录，把刚操作的业务流量保存为 HAR 文件。

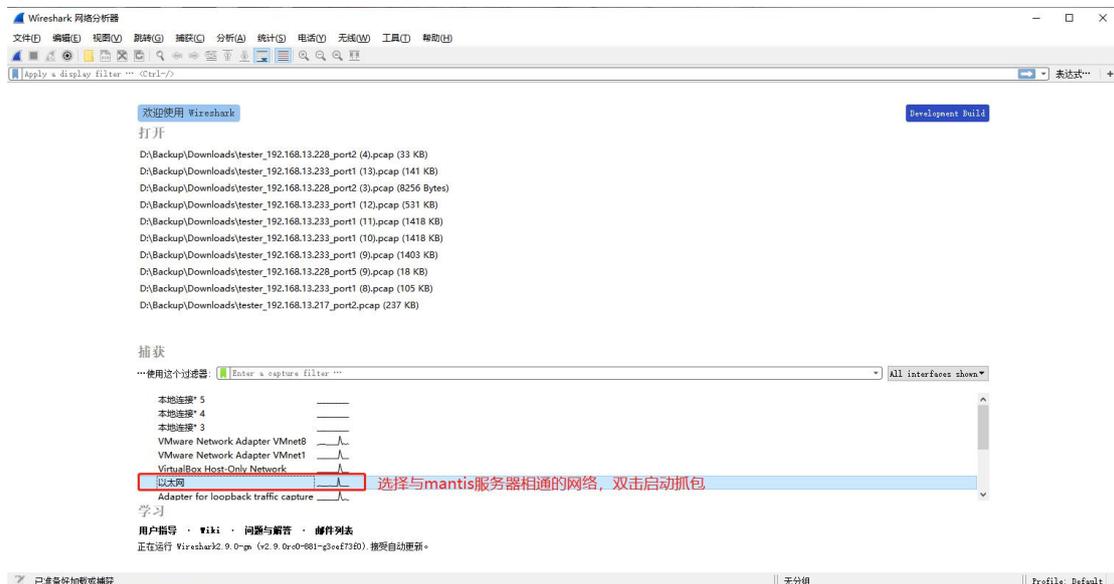


2.1.1.2 Pcap 文件方式录制事务

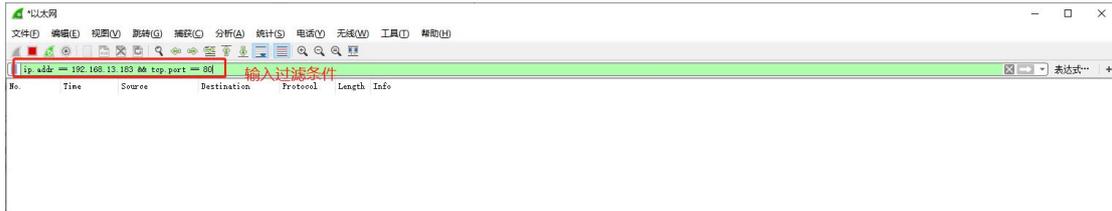
运行 Wireshark，设置过滤条件

在工作电脑上，安装抓包软件 Wireshark。如果没有安装包，可以到 Wireshark 的官网下载，这是一个免费软件，下载链接：<https://www.wireshark.org/#download>

1. 运行 Wireshark，选择工作电脑与 Mantis 服务器连接的网卡，双击启动抓包。



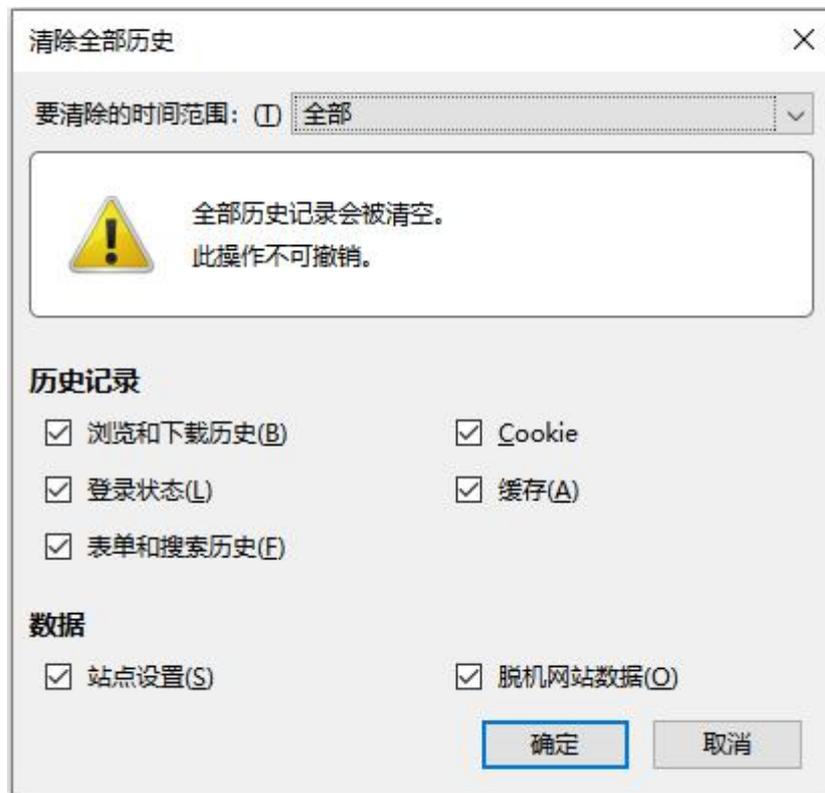
2. 因为我们只需要抓 HTTP 事务处理的报文，所以运行 Wireshark 后，要设置过滤条件，比如 Mantis 的服务器 IP 是 192.168.13.183，Web 服务的端口是 80，那么可以设置过滤条件为 `ip.addr == 192.168.13.183 && tcp.port == 80`，按回车键起效。



启动浏览器，清除缓存

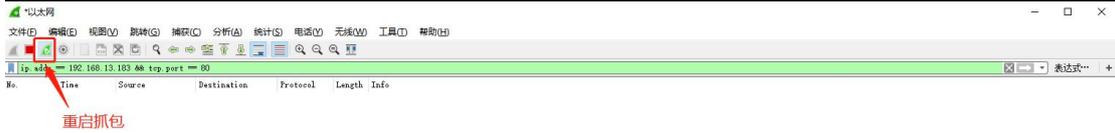
如果浏览器有缓存，就会影响 HTTP 事务处理的流程，它会优先使用缓存中的内容，而省略与服务器的通信，所以要把浏览的缓存清除干净。

各个浏览器的缓存清除快捷键都是 Ctrl + Shift + Del，打开浏览器，同时按下这三个键，选择清除选项，点击立即删除就行。

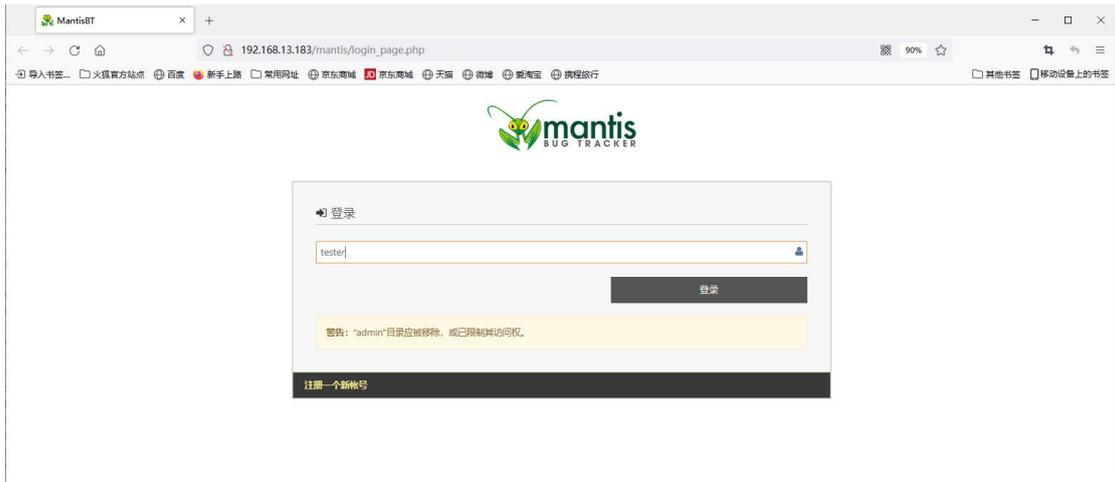


访问 Mantis, Wireshark 抓包

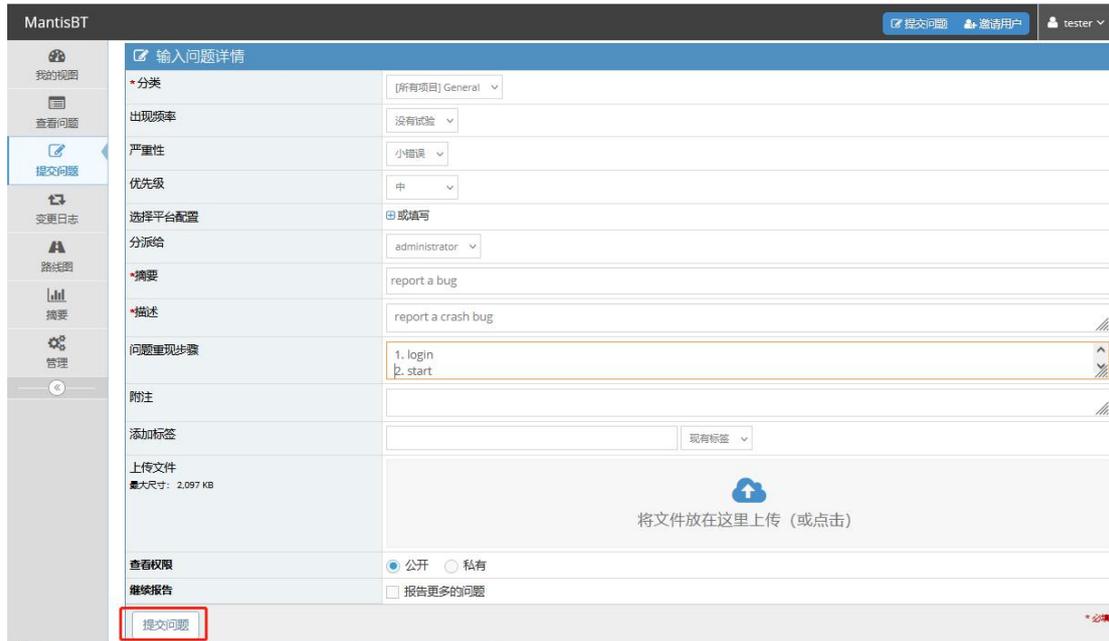
1. 在开始用浏览器访问 Mantis 之前，可以在 Wireshark 上重新启动抓包，以保证报文的干净与完整。



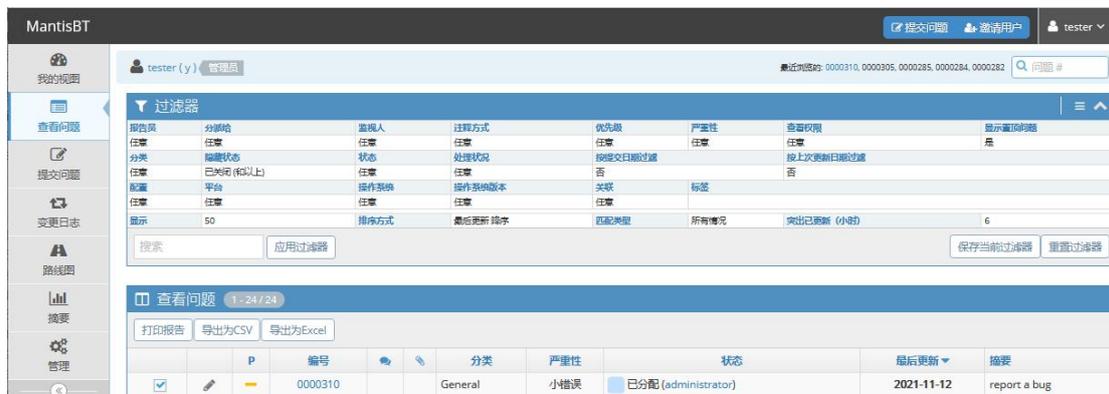
2. 与上述录制 HAR 文件过程一样，在 Firefox 浏览器中输入 mantis 服务器网址 192.168.13.183/mantis/login_page.php，用账号 tester/tester 登录系统。



- 创建并提交一个新的 bug，摘要填写“report a bug”，描述填写“report a crash bug”，问题重现步骤填写“1. login 2. start”，点击提交问题。



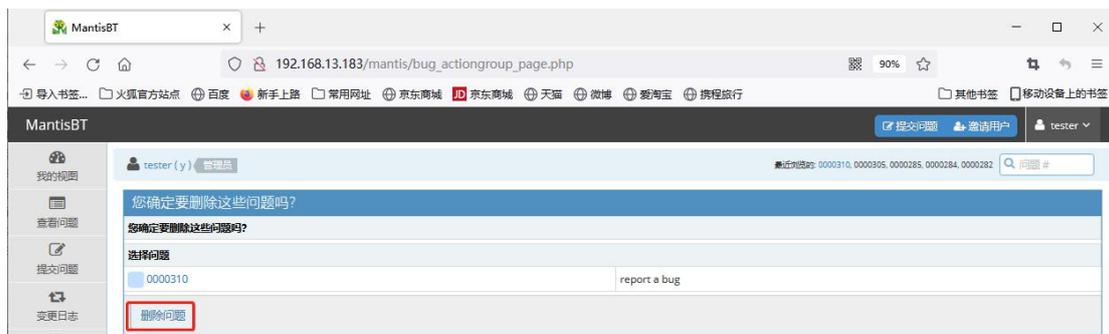
- 点击提交问题后，页面就会提示提交成功，会很快跳转到显示页面。几秒后，Mantis 自动跳转到显示页面，显示刚刚提交的 Bug。



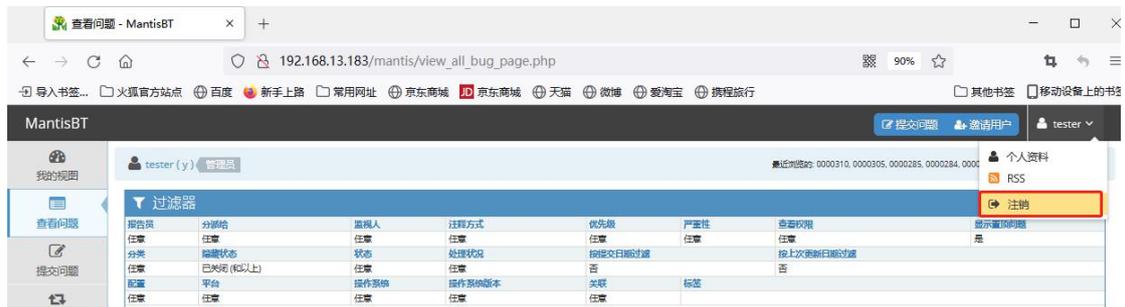
报告员	分派给	指派给	注册方式	优先级	严重性	查看权限	显示置顶问题
任意	任意	任意	任意	任意	任意	任意	是
分类	处理状态	状态	处理状况	按提交日期过滤	按上次更新日期过滤	按	
任意	已关闭(和以上)	任意	任意	否	否	否	
配置	平台	操作系统	操作系统版本	关联	标签		
任意	任意	任意	任意	任意			
显示	50	排序方式	最新更新 降序	匹配类型	所有情况	突出已更新 (小时)	6

	P	编号	分类	严重性	状态	最后更新	摘要
<input checked="" type="checkbox"/>		0000310	General	小错误	已分配 (administrator)	2021-11-12	report a bug

- 选中刚才提交的 Bug，选中“删除”选项，点击确定，删除这个 Bug。点击删除问题，Bug 将被删除，Mantis 自动跳转到 Bug 显示界面。

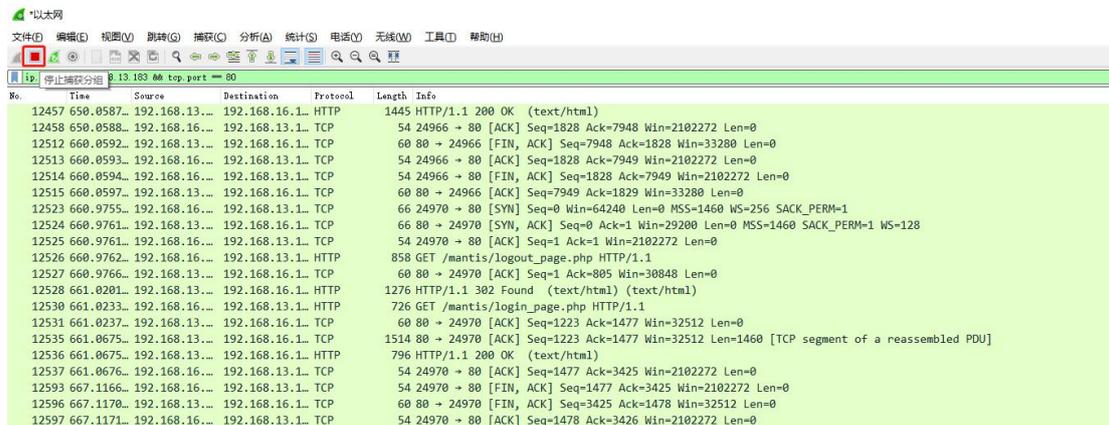


6. 点击右上角的 tester 账号的下拉菜单，点击注销，退出系统。一个完整的 HTTP 事务流程就完成了，包括 Mantis 登录，提交 Bug，删除 Bug，退出登录，关闭浏览器。

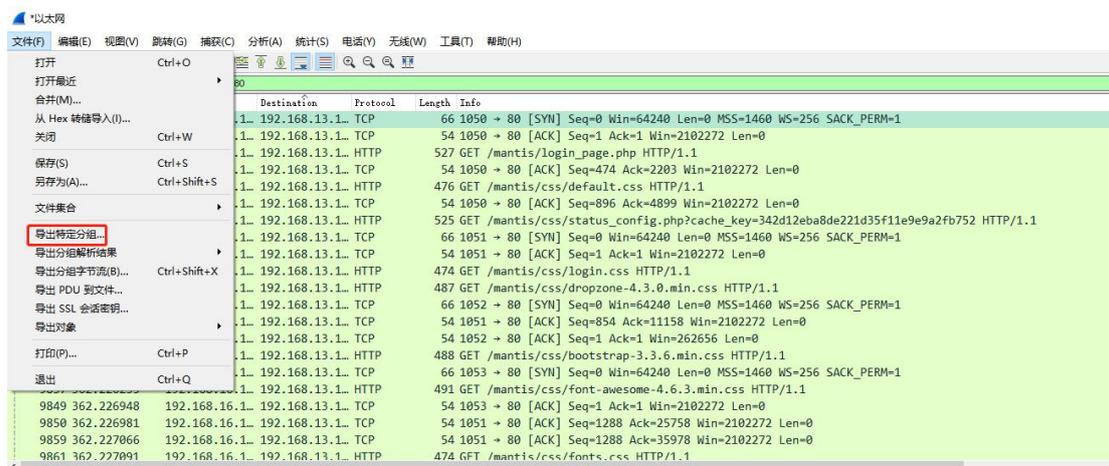


保存事务流程的抓包结果

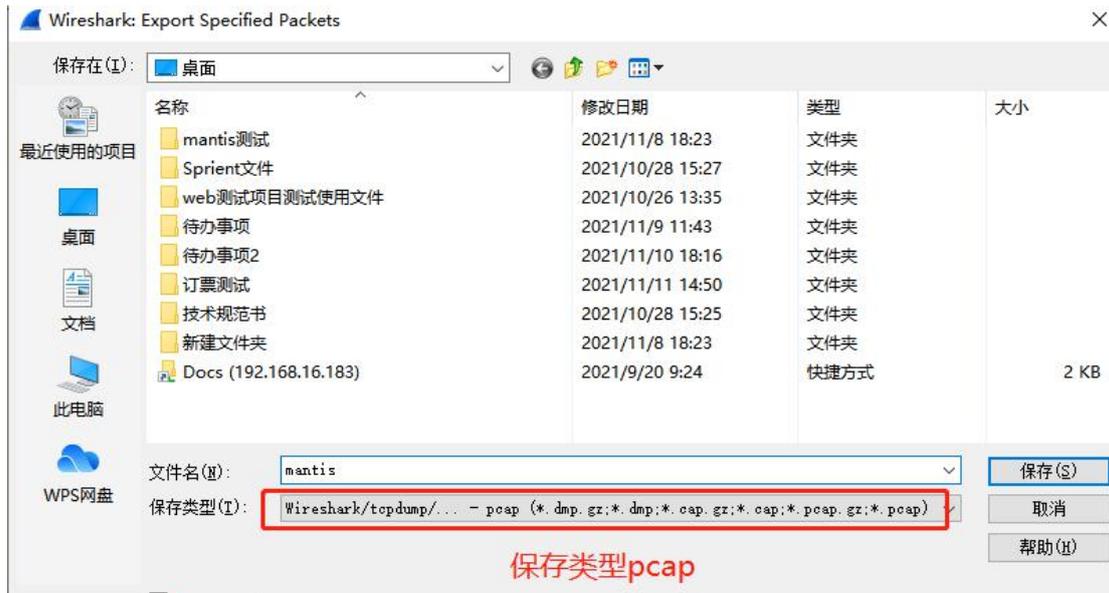
1. 点击停止捕获分组按钮，停止抓包。



2. 点击文件菜单，点击导出特定分组，把抓包结果保存。



3. 输入文件名，选择保存目录，把抓包的结果保存为 pcap 文件。



2.1.2 创建 HTTP 事务列表

1) 打开 Supernova 测试仪的 Web 界面，输入账号登录，初始账号密码 admin/admin。



2) 点击“对象”->“Web 测试项目”，创建一个 Web 测试项目，为新的 Web 测试项目命名。



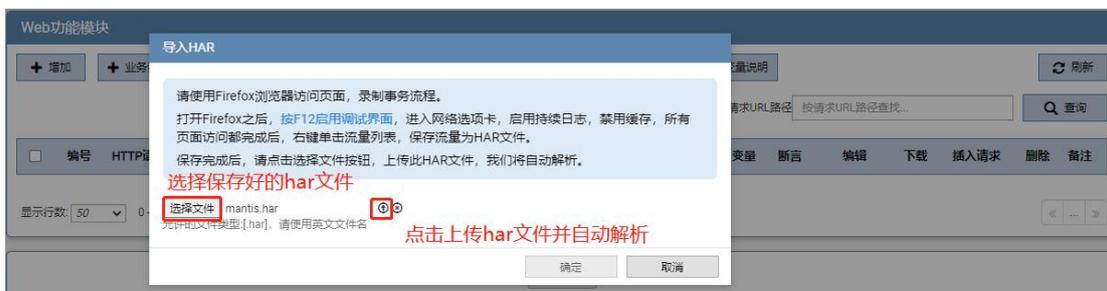
- 3) 选择新建的 Web 测试项目，在右侧窗口中创建一个 Web 功能模块来配置 HTTP 事务列表。



- 4) 输入 Web 功能模块名称，点击解析 HAR 或者解析 PCAP，就会弹出一个新的窗口。

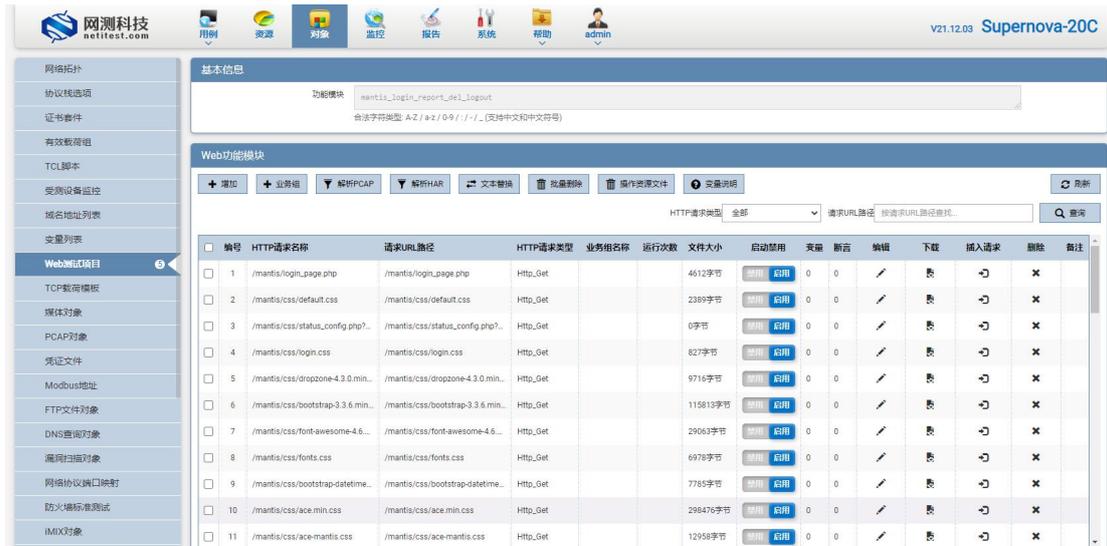


- 5) 在新的窗口中，点击浏览，选中刚刚保存的 HAR 文件或者 PCAP 文件，并点击上传按钮，将上传 HAR/PCAP 文件，系统会自动解析它，生成初始配置。解析 PCAP 与解析 HAR 生成事务列表操作方式类似，下文以上传 HAR 文件解析操作演示说明。



- 6) 如果上传解析成功，就会提示“解析 HAR 文件成功”，点击确定，就会显示 HTTP 事务的初始配置，可以删除不必要的事件，对事务过程进行精简。





编号	HTTP请求名称	请求URL路径	HTTP请求类型	业务组名称	运行次数	文件大小	启动禁用	变量	断言	编辑	下载	插入请求	删除	备注
1	/mantis/login_page.php	/mantis/login_page.php	Http_Get			4912字节	禁用 启用	0	0					
2	/mantis/css/default.css	/mantis/css/default.css	Http_Get			2389字节	禁用 启用	0	0					
3	/mantis/css/status_config.php?	/mantis/css/status_config.php?	Http_Get			0字节	禁用 启用	0	0					
4	/mantis/css/login.css	/mantis/css/login.css	Http_Get			827字节	禁用 启用	0	0					
5	/mantis/css/dropzone-4.3.0.min...	/mantis/css/dropzone-4.3.0.min...	Http_Get			9716字节	禁用 启用	0	0					
6	/mantis/css/bootstrap-3.3.6.min...	/mantis/css/bootstrap-3.3.6.min...	Http_Get			115813字节	禁用 启用	0	0					
7	/mantis/css/font-awesome-4.6...	/mantis/css/font-awesome-4.6...	Http_Get			29063字节	禁用 启用	0	0					
8	/mantis/css/fonts.css	/mantis/css/fonts.css	Http_Get			6978字节	禁用 启用	0	0					
9	/mantis/css/bootstrap-datetime...	/mantis/css/bootstrap-datetime...	Http_Get			7785字节	禁用 启用	0	0					
10	/mantis/css/ace.min.css	/mantis/css/ace.min.css	Http_Get			298476字节	禁用 启用	0	0					
11	/mantis/css/ace-mantis.css	/mantis/css/ace-mantis.css	Http_Get			12958字节	禁用 启用	0	0					

2.1.3 完善 HTTP 事务列表

完善一个简单的 HTTP 事务列表，我们需要先了解系统的变量，断言，文本替换，批量删除，操作资源文件等功能，再根据自身需求来完善 HTTP 事务列表。下面我们介绍下变量系统，断言，文本替换，批量删除和操作资源文件等功能，并完善 HTTP 事务列表。

2.1.3.1 删除资源文件

Supernova 上可以统一操作资源文件，禁用、删除、启用等。一般执行 HTTP 业务主要流程时，可以不下载非 HTML 资源文件，可以过操作资源文件删除这些资源文件。



2.1.3.2 变量参数化

在 HTTP 事务的发送信息中，如果要随机产生字符串，或者要使用特定的字符串，可以通过预先定义的变量把一些字符串放到变量中保存。预先定义变量时支持使用内部变量，内部变量可用于生成各类型随机字符串，预先定义变量时变量值可以包含内部变量。在发送信

息的内容中使用已经定义的变量,这些变量将在发送时自动替换为随机字符串或特定的字符串。

在“对象”->“变量列表”中,点击“变量说明”可以查看帮助信息,详细说明中包括如何使用变量以及支持使用的内部变量。点击“增加”点击增加,新建一个变量列表。



可以通过手动增加或解析 CSV 文件的方式来定义变量。



1) 手动增加变量

在变量页面,点击增加,新建一个变量,变量值可以包含内部变量或者字符串。

比如下面定义了变量 BugSummary,字符串格式为 8 个随机数。我们在 POST 参数里面,使用这个变量\${BugSummary},在发送 POST 请求的时候,就会把变量自动替换为“8 个随机数”,比如“xXhmuMMj”



用户定义变量之后,当前事务列表内所有的 HTTP 请求都可以引用变量。当请求内容发生变化时,只要在【变量】中更新对应变量的变量值即可,不再需要逐个修改每个请求内容。

2) 解析 csv 文件

在模拟多个用户登录系统并进行其他操作时,需要不同的用户名和密码,可以先将一组组用户名、密码信息保存到 csv 文件中(分隔符暂时只支持英文逗号),再解析到到变量中。



点击解析 CSV,弹出一个新的窗口,选择首行内容为标题行,选择文件,上传,系统会

自动解析它。

变量列表

解析CSV文件，自动解析生成多个变量

+ 增加 解析CSV 变量说明

编号	变量名称	变量类型	变量值	编辑全部变量值	编辑	删除
显示行数: 5	0 - 0 of 0					< - >

返回

导入CSV文件

编码格式: UTF-8
请选择文件编码格式

数据格式: 首行为标题行 **csv文件首行标题行**
是否将首行或者首列作为数据导入

变量名称:
为.csv文件中的每列变量值命名

分隔符: .
与.csv文件的分隔符保持一致

选择csv文件

选择文件 userinfo.csv
允许的文件类型 [.csv], 请使用英文文件名

点击上传自动解析csv文件

确定 取消

上传解析成功，就会提示“从 csv 导入并解析账号信息成功”，点击确定，就会显示到变量列表中。

导入CSV文件

编码格式: UTF-8
请选择文件编码格式

数据格式: 首行为标题行
是否将首行或者首列作为数据导入

变量名称:
为.csv文件中的每列变量值命名

分隔符: .
与.csv文件的分隔符保持一致

选择文件 userinfo.csv
允许的文件类型 [.csv], 请使用英文文件名

从csv导入并解析账号信息成功

确定 取消

变量列表

+ 增加 解析CSV

编号	变量名称	变量类型	变量值	编辑	删除
1	user	csv变量	tester/test1		✖
2	paw	csv变量	tester/test1		✖
3	BugSummary	内部变量	\$_random_digt(8)		✖

csv解析生成的变量

手动增加的变量，使用内部变量

2.1.3.2 匹配变量

当我们要从服务器的回应中提取一段内容，比如我们新建了一个 Bug，或者提交了一个订单，我们要从服务器的回应中，提取到 BugID，或者订单号，做后续删除或取消处理，可以使用匹配变量获取想要的 BugID。

在 Web 功能模块的 HTTP 列表中，选择具体的一个 HTTP 请求，点击编辑，在匹配变量选项中，在此进行变量提取。点击匹配变量旁边的问号，可以查看帮助信息。

HTTP请求文件

HTTP请求名称

合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求
响应
断言 ?
匹配变量 ?

变量1 +

变量名

变量名支持英文字母、数字和下划线_的组合 (不能以双下划线__开头), 最长为60个字节

变量占用字节

匹配变量占用的字节数, 在发送请求前进行指定字节的变量替换

正则表达式

最长为8,000个字节

默认值

若未能使用正则表达式提取出变量内容, 将使用此处设置的默认值作为该匹配变量的值, 最长为8,000个字节

补齐规则 尾部补齐

当匹配到的变量值的字节数 小于 变量占用字节数时, 使用指定的填充字节, 在值的头部/尾部补齐长度

填充字节

使用1个可见字符, 补齐内存

匹配变量

匹配变量名支持英文字母、数字和下划线_的组合 (不能以双下划线__开头), 最长为60个字节。

匹配变量的正则表达式, 是扩展的、非贪婪的PCRE正则表达式, 大小写是敏感的, 并且不能匹配多行内容, 最长8,000个字节。最终匹配到的字符串长度也不能超过8,000字节, 最多支持4个匹配分组。

匹配变量的正则表达式里面不支持使用变量。

可以使用的正则表达式说明。

元字符			
[a-z]	小写字母	[A-Z]	大写字母
[a-zA-Z]	所有字母	\d	所有数字
\w	所有数字和字母	\n	换行符
\s	空白字符	\S	非空白字符

字符匹配			
.	匹配任意单个字符	*	匹配其前面的字符任意次
*	匹配任意长度的任意字符	[]	匹配指定范围内的任意单个字符

次数匹配			
*	匹配其前字符任意次	?	匹配其前字符0次或1次
+	匹配其前字符至少1次	{m,n}	匹配其前面的字符至少m次, 至多n次

2.1.3.3 断言功能

断言内容可以包括字符串和正则表达式, 也可以直接对 HTTP 回码进行判断。在 Web 功能模块的 HTTP 列表中, 选择具体的一个 HTTP 请求, 点击编辑, 在断言选项中, 添加断言内容。

当添加多个断言内容时, 可以选择各个断言之间是与关系还是或关系。点击右边的问号, 可以查看帮助信息。

HTTP请求文件

HTTP请求名称: /mantis/login_page.php
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 **断言** 匹配变量

* 若在此事件的接收信息中, 匹配不到断言内容, 则: 匹配失败, 事务继续

各断言为 与 关系

HTTP响应断言: 无
断言统计标题:
断言内容匹配: 断言要匹配的字符串或者正则表达式的内!
若未能使用正则表达式提取出变量内容, 将使用此处设置的默认值作为该匹配变量的值, 最长为8,000个字节

在监控页面和报告中显示的统计标题。比如填写登录认证, 则统计中有登录认证断言总数, 登录认证断言成功次数两项统计

确定 取消

2.1.3.4 修改完善 HTTP 事务列表

HTTP 协议本身是无状态的, 服务器通过 cookie 辨认用户状态, Supernova 从 21.09 版本开始, 不再自动解析 Cookie, 需要通过匹配变量实现模拟 cookie 用户状态保持机制。

比如查看 Web 功能模块中的第一个 HTTP 请求响应信息的响应头中, 出现 Set-Cookie: PHPSESSID=de6518700dce88fec9bb83b218d88ef9; path=/; HttpOnly, 需要在匹配变量页签提取 PHPSESSID 的值 de6518700dce88fec9bb83b218d88ef9。可以做如下配置。

HTTP请求文件

HTTP请求名称: /mantis/login_page.php
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 断言 **匹配变量**

变量1

变量名: PHPSESSID
变量名支持英文字母、数字和下划线的组合 (不能以双下划线_开头), 最长为60个字节

变量占用字节: 32 **PHPSESSID值占用的字节**
匹配变量占用的字节数, 在发送请求前进行指定字节的变量替换

正则表达式: PHPSESSID=(\S+); **匹配非空白字符**
最长为8,000个字节

默认值: 00
若未能使用正则表达式提取出变量内容, 将使用此处设置的默认值作为该匹配变量的值, 最长为8,000个字节

补齐规则: 尾部补齐
当匹配到的变量值的字节数 小于 变量占用字节数时, 使用指定的填充字节, 在值的头部/尾部补齐长度

填充字节:
使用1个可见字符, 补齐内存

在第二个请求的请求头中 Cookie 字段使用此变量, 则发送第二个 HTTP 请求时会使用第一个请求响应中的 Set-Cookie 值。

HTTP请求文件

HTTP请求名称: /mantis/login_password_page.php
 合法字符类型: A-Z/a-z/0-9/-/_ (支持中文和中文符号)

请求 响应 断言 匹配变量

HTTP TCP

HTTP协议版本: 1.1
 HTTP协议的版本, 1.0只支持短链接, 请求之后立即关闭连接; 1.1支持长连接, 即多个请求一个TCP连接; 2.0支持长连接下头部压缩、二进制分帧、多路复用

请求URL路径: /mantis/login_password_page.php
 文件要保存的绝对路径, 不能为空, 必须以/开头, 比如 / 或者 /news/sports/

请求URL名称类型: 固定
 请求URL路径名称命名方式, 固定是保持请求URL路径名称不变, 可变量是请求URL路径名称可变的

Host字段策略: 自动填充
 自动填充时, Host字段自动填充为服务器IP地址; 指定地址时, 可以指定固定的IP地址或者域名

HTTP请求头:


```

US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.13.183
Connection: keep-alive
Referer: http://192.168.13.183/mantis/login_page.php
Cookie: PHPSESSID=${PHPSESSID}
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
        
```

 * 如果POST的数据是json格式, 请在请求头中添加Content-Type: application/json
 * 如果POST的数据是简单的文本, 请在请求头中添加Content-Type: application/x-www-form-urlencoded
 * 可以使用关联的变量列表中的变量, 比如\${user}
 * 如果POST请求格式是Multipart参数, 请在请求头中添加Content-Type: multipart/form-data; boundary=自定义boundary, 例如: Content-Type: multipart/form-data; boundary=netitest123456

确定 取消

其他请求中的 Cookie 字段都需如此配置, 在此不一一罗列。

因为 HTTP 的事务的推动, 大多是用 POST 方式, 由浏览器向服务器提交数据触发。所以, 我们先查询出所有的 POST 请求。可以看出总共有 5 个 POST 请求, 1、2 是 Mantis 系统登录, 3 是提交 Bug, 4、5 是删除和确认。

Web功能模块

HTTP请求类型: 全部 请求URL路径: 按请求URL路径查找...

编号	HTTP请求名称	请求URL路径	HTTP请求类型	业务组名称	运行次数	文件大小	启动禁用	支持	通过HTTP请求类型进行筛选	下载	插入请求	删除	备注
1	/mantis/login_page.php	/mantis/login_page.php	Http_Get			4612字节	禁用	启用	0	0	启用	删除	×
2	/mantis/login_password_page.php	/mantis/login_password_page.php	Http_Post			5018字节	禁用	启用	0	0	启用	删除	×
3	/mantis/login.php	/mantis/login.php	Http_Post			0字节	禁用	启用	0	0	启用	删除	×
4	/mantis/login_cookie_test.php?refu...	/mantis/login_cookie_test.php?refu...	Http_Get			0字节	禁用	启用	0	0	启用	删除	×
5	/mantis/index.php	/mantis/index.php	Http_Get			0字节	禁用	启用	0	0	启用	删除	×
6	/mantis/my_view_page.php	/mantis/my_view_page.php	Http_Get			35151字节	禁用	启用	0	0	启用	删除	×
7	/mantis/bug_report_page.php	/mantis/bug_report_page.php	Http_Get			15712字节	禁用	启用	0	0	启用	删除	×
8	/mantis/bug_report.php?posted=1	/mantis/bug_report.php?posted=1	Http_Post			8708字节	禁用	启用	0	0	启用	删除	×
9	/mantis/view_all_bug_page.php	/mantis/view_all_bug_page.php	Http_Get			41845字节	禁用	启用	0	0	启用	删除	×
10	/mantis/bug_actiongroup_page.php	/mantis/bug_actiongroup_page.php	Http_Post			9486字节	禁用	启用	0	0	启用	删除	×
11	/mantis/bug_actiongroup_nho	/mantis/bug_actiongroup_nho	Http_Post			0字节	禁用	启用	0	0	启用	删除	×

Web功能模块

HTTP请求类型: 请求URL:

编号	HTTP请求名称	请求URL路径	HTTP请求类型	业务组名称	运行次数	文件大小	启动禁用	变量	断言	编辑	下载	插入请求	删除	备注
1	/mantis/login_password_page.php	/mantis/login_password_page.php	Http_Post	登录		5018字节	<input type="button" value="禁用"/> <input type="button" value="启用"/>	0	0	<input type="button" value="✎"/>	<input type="button" value="📄"/>	<input type="button" value="➡"/>	<input type="button" value="✖"/>	
2	/mantis/login.php	/mantis/login.php	Http_Post			0字节	<input type="button" value="禁用"/> <input type="button" value="启用"/>	0	0	<input type="button" value="✎"/>	<input type="button" value="📄"/>	<input type="button" value="➡"/>	<input type="button" value="✖"/>	
3	/mantis/bug_report.php?posted=1	/mantis/bug_report.php?posted=1	Http_Post	提交bug		8708字节	<input type="button" value="禁用"/> <input type="button" value="启用"/>	0	0	<input type="button" value="✎"/>	<input type="button" value="📄"/>	<input type="button" value="➡"/>	<input type="button" value="✖"/>	
4	/mantis/bug_actiongroup_page.php	/mantis/bug_actiongroup_page.php	Http_Post			9486字节	<input type="button" value="禁用"/> <input type="button" value="启用"/>	0	0	<input type="button" value="✎"/>	<input type="button" value="📄"/>	<input type="button" value="➡"/>	<input type="button" value="✖"/>	
5	/mantis/bug_actiongroup.php	/mantis/bug_actiongroup.php	Http_Post	删除bug		0字节	<input type="button" value="禁用"/> <input type="button" value="启用"/>	0	0	<input type="button" value="✎"/>	<input type="button" value="📄"/>	<input type="button" value="➡"/>	<input type="button" value="✖"/>	

显示行数: 1 - 5 of 5

下面我们举例修改下 HTTP 事务列表，说明变量的引用和匹配变量的使用方式：

1) 点击第一个 POST 请求的编辑按钮，点击编辑进入请求编辑页面。在请求信息中，使用预先定义的变量 `user` 来替换用户名。变量名 `user` 对应两个变量值，则每个虚拟用户在发送信息时，将循环使用这两个变量值进行发送。

HTTP请求文件

HTTP请求名称:

合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

自动填充时，POST字段自动填充为服务器端IP地址；指定地址时，可以指定固定的IP地址或域名

HTTP请求头

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.13.183
```

* 如果POST的数据是json格式，请在请求头中添加Content-Type: application/json
 * 如果POST的数据是简单的文本，请在请求头中添加Content-Type: application/x-www-form-urlencoded
 * 可以使用关联的变量列表中的变量，比如\$(user)
 * 如果POST请求格式是Multipart参数，请在请求头中添加Content-Type: multipart/form-data; boundary=自定义boundary.
 例如: Content-Type: multipart/form-data; boundary=netitest123456

HTTP请求类型:

HTTP请求的方式: Get只请求页面, Post可以在请求页面时, 提交数据

POST请求格式:

选择Post内容的格式, 对Post内容进行编辑

动作提交数据: **使用变量user**

2) 点击第二个 POST 请求的编辑按钮，点击编辑进入请求编辑页面。在请求信息中，使用预先定义的变量 `user` 和 `paw` 来替换用户名、密码。变量名 `user` 和 `paw` 分别对应两个变量值，每个虚拟用户在发送信息时，将循环使用这些变量值进行发送。具体发送报文信息可以看 2.4 章节中查看登录时用户名、密码。

HTTP请求文件

HTTP请求名称

合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 断言 匹配变量

HTTP TCP

Host字段策略

Host字段策略说明: 自动填充时, Host字段自动填充为服务器IP地址; 指定地址时, 可以指定固定的IP地址或者域名

HTTP请求头

```

    User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/4.0
    Accept-Encoding: gzip, deflate
    Content-Type: application/x-www-form-urlencoded
    Origin: http://192.168.13.183
    Connection: keep-alive
    Referer: http://192.168.13.183/mantis/login_password_page.php
    Cookie: PHPSESSID=de651870dce88fec9bb83b218d88ef9
    Upgrade-Insecure-Requests: 1
    Pragma: no-cache
    Cache-Control: no-cache
    
```

* 如果POST的数据是json格式, 请在请求头中添加Content-Type: application/json
 * 如果POST的数据是简单的文本, 请在请求头中添加Content-Type: application/x-www-form-urlencoded
 * 可以使用关联的变量列表中的变量, 比如\$(user)
 * 如果POST请求格式是Multipart参数, 请在请求头中添加Content-Type: multipart/form-data; boundary=自定义boundary, 例如: Content-Type: multipart/form-data; boundary=netitest123456

HTTP请求类型

HTTP请求的方式, Get只请求页面, Post可以在请求页面时, 提交数据

POST请求格式

选择Post内容的格式, 对Post内容进行编辑

动作提交数据

用户名和密码使用变量替换

3) 点击第三个 POST 请求的编辑按钮, 点击编辑进入请求编辑页面。在请求信息的参数里使用预先定义的变量 BugSummary 用来区分和辨识每个 Bug, 具体发送报文信息可以查看 2.4 章节查看提交 Bug 摘要信息。

HTTP请求文件

HTTP请求名称

合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 断言 匹配变量

HTTP TCP

POST请求格式

选择Post内容的格式, 对Post内容进行编辑

Multipart参数

名称	数值	操作
bug_report_token	20211112awnvb0sQ3Vc	+
m_id	0	-
project_id	1	-
category_id	1	-
reproducibility	70	-
severity	50	-
priority	30	-
platform		-
os		-
os_build		-
handler_id	1	-
summary	t a bug \$(BugSummary)	-
description	report a crash bug	-

4) 点击第三个 POST 请求, 从响应信息中匹配变量, 把提交 Bug 的 BugID 提取出来,

在后面删除的时候使用。进入匹配变量选项，定义一个变量名 ReportID，正则表达式为 `view.php?id=(0-9+)\\" title=\"[^\"]+ report a bug \\w+`，主要是提取 Bug 的摘要信息，以及 BugID，在后面删除的时候会用到。

HTTP请求文件

HTTP请求名称: /mantis/bug_report.php?posted=1
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 断言 匹配变量

变量1

变量名: ReportID
变量名支持英文字母、数字和下划线的组合 (不能以双下划线_开头), 最长为60个字节

变量占用字节: 7 BugID所占字节长度
匹配变量占用的字节数, 在发送请求前进行指定字节的变量替换

正则表达式: view.php?id=([0-9]+)\" title=\"[^\"]+ report a
最长为8,000个字节

默认值: 0000
若未能使用正则表达式提取出变量内容, 将使用此处设置的默认值作为该匹配变量的值, 最长为8,000个字节

补齐规则: 头部补齐
当匹配到的变量值的字节数 小于 变量占用字节数时, 使用指定的填充字符, 在值的头部/尾部补齐长度

填充字符: 0 BugID结构前边为0, 后面为编号
使用1个可见字符, 补齐内存

确定 取消

5) 在断言选项中, 可以添加断言, 断言内容: `view.php?id=(0-9+)\\" title=\"[^\"]+ report a bug \\w+`, 包含正则表达式, 如果匹配不到断言内容, 则匹配失败事务中断。

HTTP请求文件

HTTP请求名称: /mantis/bug_report.php?posted=1
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

请求 响应 断言 匹配变量

* 若在此事件的接收信息中, 匹配不到断言内容, 则: 匹配失败, 事务中断 各断言为 与 关系

HTTP响应断言: 正则表达式匹配
在HTTP的响应中, 匹配固定字符串, 或进行正则表达式匹配

断言统计标题: BugID
在监控页面和报告中显示的统计标题, 比如填充认证, 则统计中有登录认证断言总数, 登录认证成功次数两项统计

断言内容匹配: view.php?id=([0-9]+)\" title=\"[^\"]+ report a
若未能使用正则表达式提取出变量内容, 将使用此处设置的默认值作为该匹配变量的值, 最长为8,000个字节

确定 取消

6) 再重新查找出所有的 POST 事件, 编辑 POST 请求 4, 把 POST 参数中的 BugID (比如 53946), 换成提取变量 `${ReportID}`, 对应提取到的变量值。具体发送报文信息可以看 2.4 章节中按 BugID 删除 Bug。

HTTP请求名称: /mantis/bug_actiongroup_page.php
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

Host字段策略: 自动填充
自动填充时, Host字段自动填充为服务器IP地址; 指定地址时, 可以指定固定的IP地址或者域名

HTTP请求头: Referer: http://192.168.13.183/mantis/view_all_bug_page.php
Cookie: PHPSESSID=de6518700dce88fec9bb83b218d88ef9;
MANTIS_secure_session=1;
MANTIS_STRING_COOKIE=OynP5dTnfeVeaJjwFkUmksXURFar4b4rZBT2-kQgBZ0atU2XPPHExHnnaA7I88bb;
MANTIS_BUG_LIST_COOKIE=305%2C304%2C303%2C302%2C301%2C300%2C299%2C298%2C297%2C296%2C295%2C294%2C293%2C292%2C291%2C290%2C289%2C288%2C287%2C286
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

HTTP请求类型: Http_Post
HTTP请求的方式, Get只请求页面, Post可以在请求页面时, 提交数据

POST请求格式: Body编辑
选择Post内容的格式, 对Post内容进行编辑

动作提交数据: bug_arr%5B%5D=\${ReportID}&action=DELETE

7) 编辑 POST 请求 5, 把 POST 参数中的 BugID (比如 53946), 换成提取变量\${ReportID}, 对应提取到的变量值。

HTTP请求名称: /mantis/bug_actiongroup.php
合法字符类型: A-Z / a-z / 0-9 / : / - / _ (支持中文和中文符号)

Host字段策略: 自动填充
自动填充时, Host字段自动填充为服务器IP地址; 指定地址时, 可以指定固定的IP地址或者域名

HTTP请求头: Referer: http://192.168.13.183/mantis/bug_actiongroup_page.php
Cookie: PHPSESSID=de6518700dce88fec9bb83b218d88ef9;
MANTIS_secure_session=1;
MANTIS_STRING_COOKIE=OynP5dTnfeVeaJjwFkUmksXURFar4b4rZBT2-kQgBZ0atU2XPPHExHnnaA7I88bb;
MANTIS_BUG_LIST_COOKIE=305%2C304%2C303%2C302%2C301%2C300%2C299%2C298%2C297%2C296%2C295%2C294%2C293%2C292%2C291%2C290%2C289%2C288%2C287%2C286
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

HTTP请求类型: Http_Post
HTTP请求的方式, Get只请求页面, Post可以在请求页面时, 提交数据

POST请求格式: Body编辑
选择Post内容的格式, 对Post内容进行编辑

动作提交数据: bug_actiongroup_DELETE_token=202111120zk5c5-w828mn4xiPaq1-vCOEYPR2iW8&action=DELETE&bug_arr%5B%5D=\${ReportID}

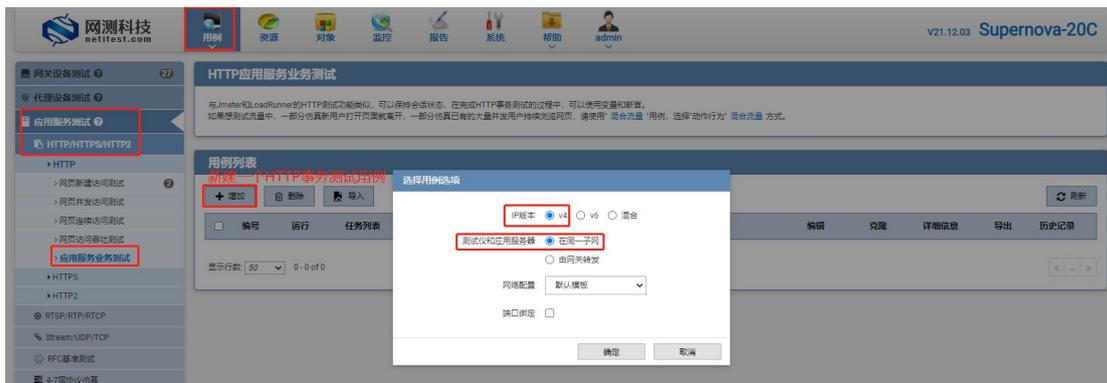
以上步骤举例了在变量列表中如何使用变量及匹配变量, 在实际使用过程中, 可能还会遇到 Session 或 Token 等情况, 如提交 bug 参数中用到的 bug_report_token 以及删除 bug 时用到

的 bug_actiongroup_DELETE_token，需要在此动作的前一个动作或某个动作的响应信息中进行匹配变量的设置，提取相关内容。

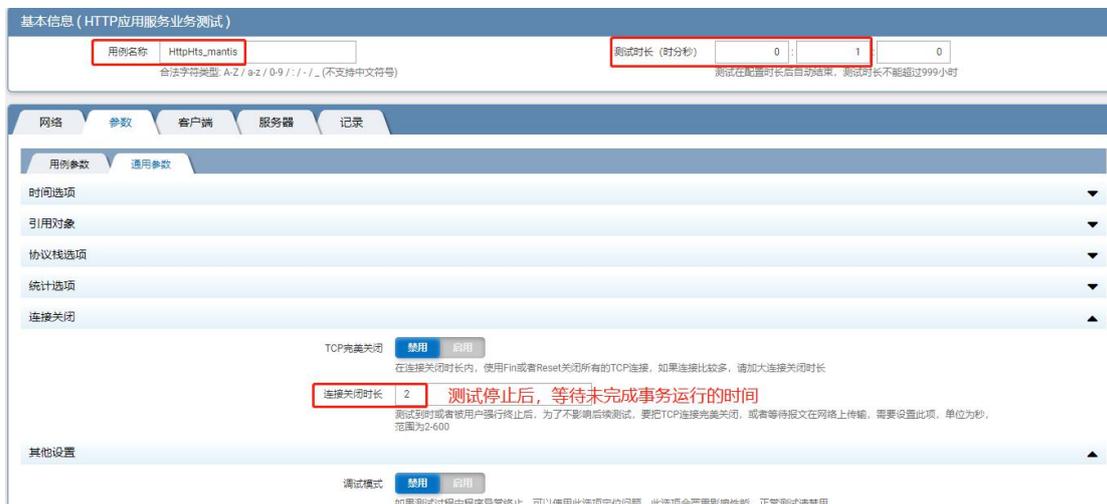
2.2 创建事务用例

创建一个 HTTP 应用服务业务测试用例，引用上述创建的事务列表，并修改配置参数，比如端口的 IP 地址以及虚拟用户的数量。

1) 依次点击，用例 -> 应用服务测试 -> HTTP/HTTPS/HTTP2 -> HTTP ->应用服务业务测试，点击“增加”，就会弹出增加 HTTP 应用服务业务测试用例的对话框。因为测试仪的 port1 端口与 Mantis 服务器的 eth1 端口是光纤直连，所以选同一子网。



2) 输入 HTTP 应用服务业务测试用例的名称，设置测试时长和等待终止时长（测试时间结束之后，等待未完成事务正常运行结束的时间）。



3) 根据网络拓扑和 IP 设置，配置端口和 IP 地址。

基本信息 (HTTP应用服务业务测试)

用例名称: HttpHts_mantis 测试时长 (时:分:秒): 0 : 1 : 0
合法字符类型: A-Z / a-z / 0-9 / - / _ (不支持中文字符) 测试在配置时长后自动结束, 测试时长不能超过999小时

网络 参数 客户端 服务器 记录

保存网络设置 配置端口绑定 选择port1

客户端 port1 禁用端口 MAC: 68:91:d0:63:4c:bc

网络配置 CPU核绑定 限速 抓包 网卡设置

IPv4虚拟主机 IPv4边界网关 QinQ

IPv4虚拟主机 + 添加虚拟主机网 移除虚拟主机网 禁用虚拟主机网

第 1 个虚拟主机网

虚拟主机网主机地址或范围: 17.1.2-17.1.2.201 **测试仪模拟的客户端IP地址范围**

虚拟主机网主机地址范围, 每个IP地址模拟一个主机。启用虚拟网络时, 与各个虚拟网络的网络地址或, 模拟更大范围的虚拟主机群。

虚拟主机网掩码: 16

虚拟主机网掩码长度, IPv4范围0-32, IPv6范围0-128, 启用虚拟网络时, 虚拟主机网掩码不能小于虚拟网络区域的网络掩码

虚拟主机网网关: 17.1.1.1

虚拟主机网的下一跳网关, 禁用虚拟网络时有效, 启用时在虚拟网络中配置。

虚拟主机网VLAN: 1

虚拟主机网VLAN, 禁用虚拟网络时有效, 启用时在虚拟网络中配置。

虚拟主机模拟角色: 只模拟客户端

虚拟主机测试过程中模拟的角色, 可以选择只模拟客户端/只模拟服务器/同时模拟客户端和服务端

服务器IP地址或范围: 17.1.1.100 **与port1相连的服务器eth1网口IP地址**

4) 在参数->用例参数中, 配置虚拟用户数量、请求超时秒数、HTTP 请求 hash 表实体数量参数。

基本信息 (HTTP应用服务业务测试)

用例名称: HttpHts_mantis 测试时长 (时:分:秒): 0 : 1 : 0
合法字符类型: A-Z / a-z / 0-9 / - / _ (不支持中文字符) 测试在配置时长后自动结束, 测试时长不能超过999小时

网络 参数 客户端 服务器 记录

用例参数 通用参数

用户admin占用内存: 12 GB
用户admin申请占用的内存数量, 可以到资源->内存管理页面申请

用例运行占用内存: 12 GB
运行此用例占用的内存空间, 包括OPDK的大页内存, 以及SSL会话占用的普通系统内存, 范围: 2-12GB

DPDK大页内存占比: 60 %
DPDK大页内存占用例运行内存的百分比, 运行HTTPS并发用例时, 建议大页内存占比50%, 剩余内存用于建立SSL会话, 范围: 10%-95%

虚拟用户数量: 50
范围: 1-32768, 每个客户端网口和CPU核, 至少要有个虚拟用户

循环次数: 0
每个虚拟用户执行Web功能模块的次数, 0表示无限制

请求超时秒数: 10
发出HTTP请求后, 等待响应的超时秒数, 范围: 1-300

HTTP请求hash表实体数量: 1000
hash表内允许存放的HTTP请求对象的数量, 范围: 16-2^30

- 虚拟用户数量: 测试仪客户端同时模拟多少虚拟用户进行并发测试;
- 请求超时秒数: 发出 HTTP 请求后, 等待响应的超时秒数;
- HTTP 请求 hash 表实体数量: 每一个 hash 实体代表一条不同的 HTTP 请求, HTTP 请求 hash 表实体数量=虚拟用户数*HTTP 请求数量, 比如 2.1 章节中配置的事务列表中有 14 个请求, 每个请求中配置有变量, 当变量长度固定的话, 每个请求所占长度也一致, 分配 hash 表实体数量时: hash 表实体数量=虚拟用户数量*14; 上述配置中用户名和密码变量是有两个长度, 且两个变量值的长度是一样的, 则有两种组合, 登录的两个请求中实际上是由两种组合, 分配 hash 表实体数量时: hash 表实体数量=虚拟用户数量*16, 可适当多配置一些。

5) 在客户端-中, 选择要运行的 Web 功能模块即事务列表, 以及定义而变量列表。点击

保存，保存 HTTP 应用服务业务测试用例的配置。



因为本文档中的 HTTP 服务是默认的 80 端口，可不修改服务器端口，若 HTTP 服务使用的其他端口，则需在服务器选项中配置服务器端口。



2.3 运行事务用例

1) 测试用例配置完成之后，点击运行启动刚刚保存的 HTTP 应用服务业务测试用例。

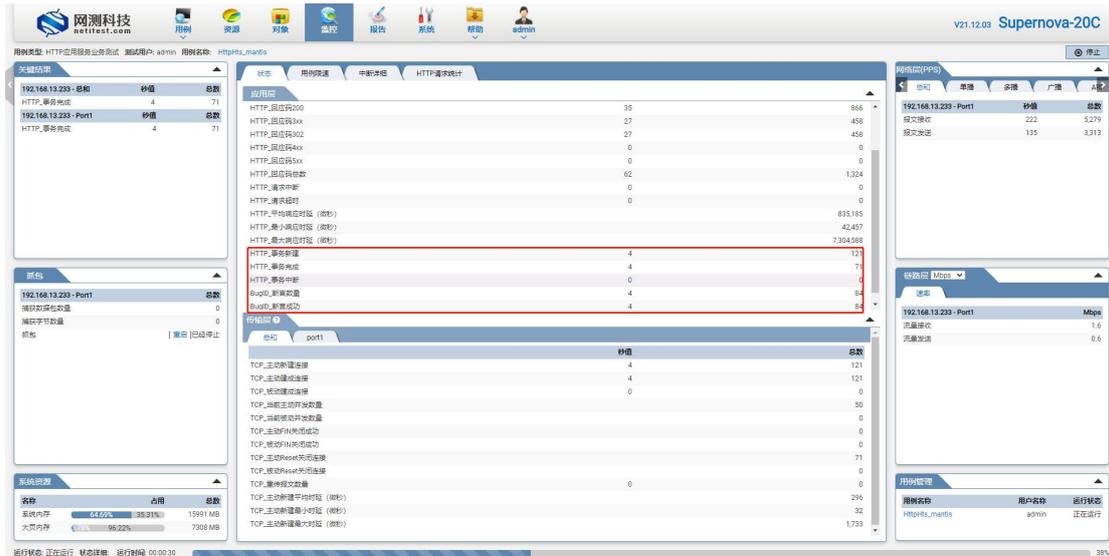


2) 用例启动后自动跳转监控页面，监控页面实时显示测试数据。

如果网络配置正确，测试用例正常启动后，就会显示运行状态。主要的状态数据有两个。

- ✧ HTTP_事务完成：显示的是上一秒完成和总共完成的 HTTP 事务数量。
- ✧ HTTP_事务中断：显示的是上一秒中断和总共中断的 HTTP 事务数量。HTTP 事务中断的原因有很多，比如我们提交 Bug 太快，Mantis 服务器并没有返回新建的 BugID，导致我们断言的时候匹配不到断言内容，事务中断。如果有中断，中断的具体信息可以在中断详细页面看到。

另外由于事务列表中配置了断言，还会统计断言成功的状态。



详细统计显示

可以实时看到每个事件发送的样本数，时延最小最大值和平均值，请求完成速率等等。运行结束后您也可以通过查看关键结果来查看详细统计信息。



事务名称	运行次数	运行速率	成功次数	成功速率	出错次数	平均模块完成时间(ms)	最小模块完成时间(ms)	最大模块完成时间(ms)
mantis_login_report_del_logout	264	4.4/sec	214	3.6/sec	0	12642	2023	17976

事务名称	请求路径	HTTP请求次数	HTTP请求速率	HTTP响应次数	HTTP响应速率	断言失败次数	HTTP出错次数	HTTP出错百分比	发送速率(KB/sec)	接收速率(KB/sec)	平均响应时延(毫秒)	最小响应时延(毫秒)	最大响应时延(毫秒)
mantis_login_report_del_logout	/mantis/login_page.php	262	4.4/sec	262	4.4/sec	0	0	0.00%	1.97	9.59	965195	43533	49552
mantis_login_report_del_logout	/mantis/login_password_page.php	258	4.3/sec	258	4.3/sec	0	0	0.00%	3.00	9.32	434925	43315	7272
mantis_login_report_del_logout	/mantis/login.php	258	4.3/sec	258	4.3/sec	0	0	0.00%	3.13	3.87	643506	141162	15100
mantis_login_report_del_logout	/mantis/login_cookie_test.php?ret...	249	4.2/sec	249	4.2/sec	0	0	0.00%	2.91	2.90	345327	75229	5431
mantis_login_report_del_logout	/mantis/index.php	249	4.2/sec	249	4.2/sec	0	0	0.00%	2.79	2.93	417106	64066	6031
mantis_login_report_del_logout	/mantis/my_view_page.php	249	4.2/sec	249	4.2/sec	0	0	0.00%	2.82	22.54	744920	145046	10671
mantis_login_report_del_logout	/mantis/bug_report_page.php	249	4.2/sec	249	4.2/sec	0	0	0.00%	2.81	22.96	509966	79082	7994
mantis_login_report_del_logout	/mantis/bug_report.php?posted=1	247	4.1/sec	232	3.9/sec	0	0	0.00%	13.11	13.03	538479	334188	80794
mantis_login_report_del_logout	/mantis/view_all_bug_page.php	225	3.8/sec	225	3.8/sec	0	0	0.00%	2.33	31.61	553243	105470	7534
mantis_login_report_del_logout	/mantis/bug_actiongroup_page.php	222	3.7/sec	222	3.7/sec	0	0	0.00%	4.14	13.52	492221	117809	6862
mantis_login_report_del_logout	/mantis/bug_actiongroup.php	222	3.7/sec	222	3.7/sec	0	0	0.00%	4.39	5.64	687339	232457	15382
mantis_login_report_del_logout	/mantis/view_all_bug_page.php	218	3.6/sec	218	3.6/sec	0	0	0.00%	3.57	30.77	554690	105209	9714
mantis_login_report_del_logout	/mantis/bug_report_page.php	216	3.6/sec	216	3.6/sec	0	0	0.00%	3.51	4.38	387935	42457	5774
mantis_login_report_del_logout	/mantis/login_page.php	214	3.6/sec	214	3.6/sec	0	0	0.00%	3.01	7.83	443021	46475	6192

3) 查看下载测试报告

点击报告，进入报告查看页面，可以查看测试结果，导出 PDF、Word 等格式的测试报告。

找到刚刚运行的用例，打开测试结果。



点击生成报告就会生成摘要和图表数据。摘要和图表生成 100%时，点击关闭可查看报告摘要数据和图表信息，继续点击生成文档，可以生成 HTML、PDF、Word、Excel 格式的报

告。点击下载 PDF、下载 HTML、下载 Word、下载 Excel 就可以把报告保存到本地电脑。



2.4 抓包查看报文

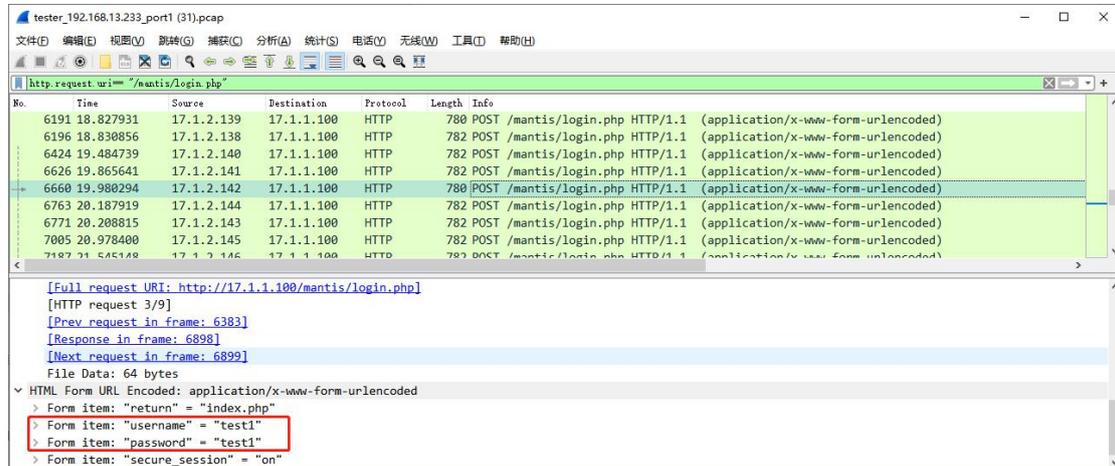
在创建事务用例，配置网络时，设置报文捕获。



运行结束后，在监测页面下载 pcap 包，使用 Wireshark 打开。

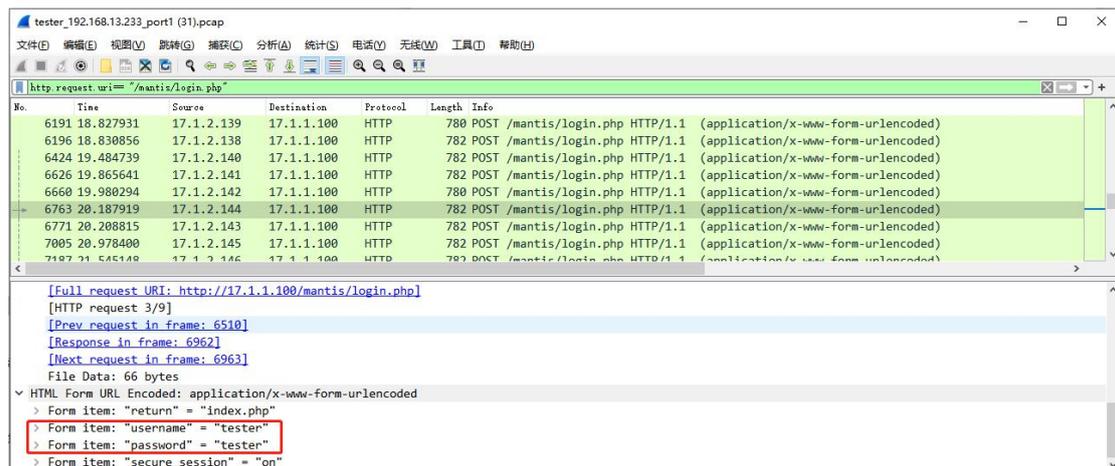
1) 查看登录时用户名、密码

在过滤器输入 `http.request.uri== "/mantis/login.php"`, 过滤出登录页面, 查看参数信息, 变量名 `user` 和 `paw` 分别对应两个变量值, 报文中用户名、密码循环使用这两个不同的变量值。



Wireshark capture showing a POST request to `/mantis/login.php`. The request body is HTML Form URL Encoded. The form items are:

- Form item: "return" = "index.php"
- Form item: "username" = "test1"
- Form item: "password" = "test1"
- Form item: "secure_session" = "on"

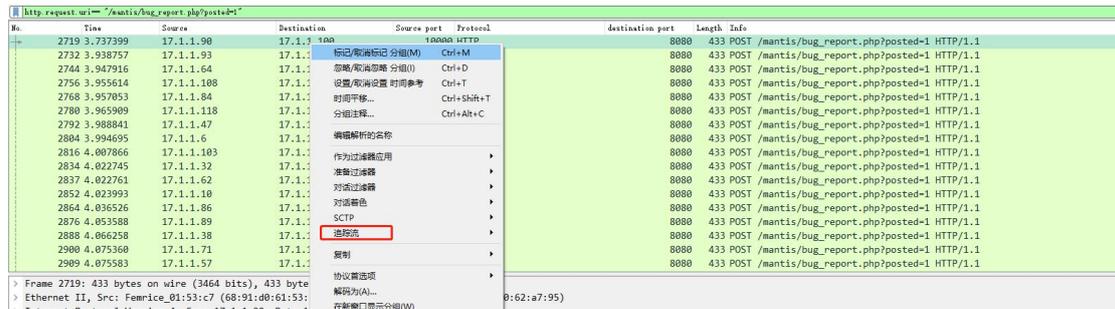


Wireshark capture showing a POST request to `/mantis/login.php`. The request body is HTML Form URL Encoded. The form items are:

- Form item: "return" = "index.php"
- Form item: "username" = "tester"
- Form item: "password" = "tester"
- Form item: "secure_session" = "on"

2) 查看提交 Bug 摘要信息

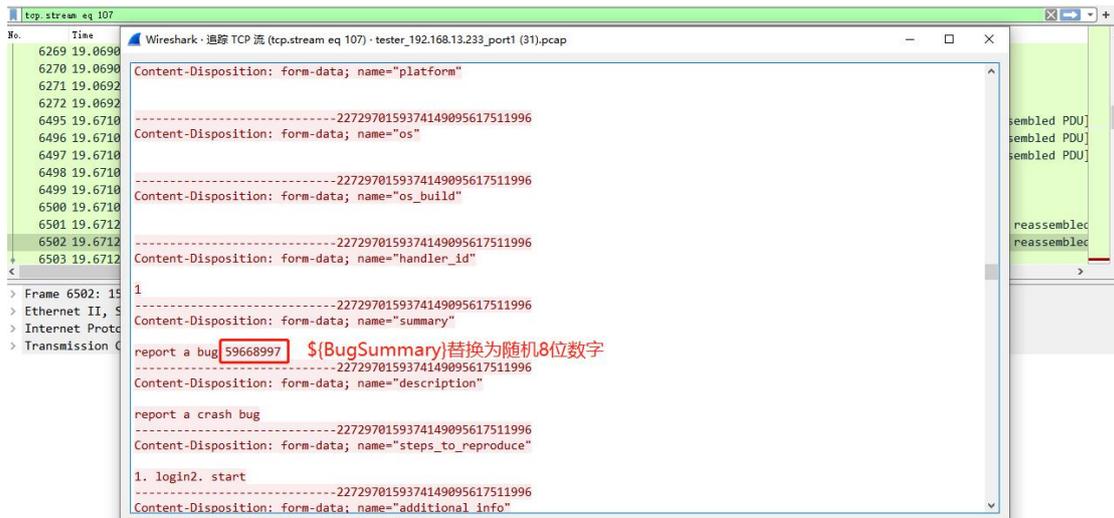
在过滤器输入 `http.request.uri== "/mantis/bug_report.php?posted=1"`, 过滤出提交 Bug 页面, 选中一个跟踪 TCP 流, 查看报文发送中的 `summary` 信息, `summary` 已被 `BugSummary` 变量值替换。



Wireshark capture showing a POST request to `/mantis/bug_report.php?posted=1`. The request body is HTML Form URL Encoded. The form items include:

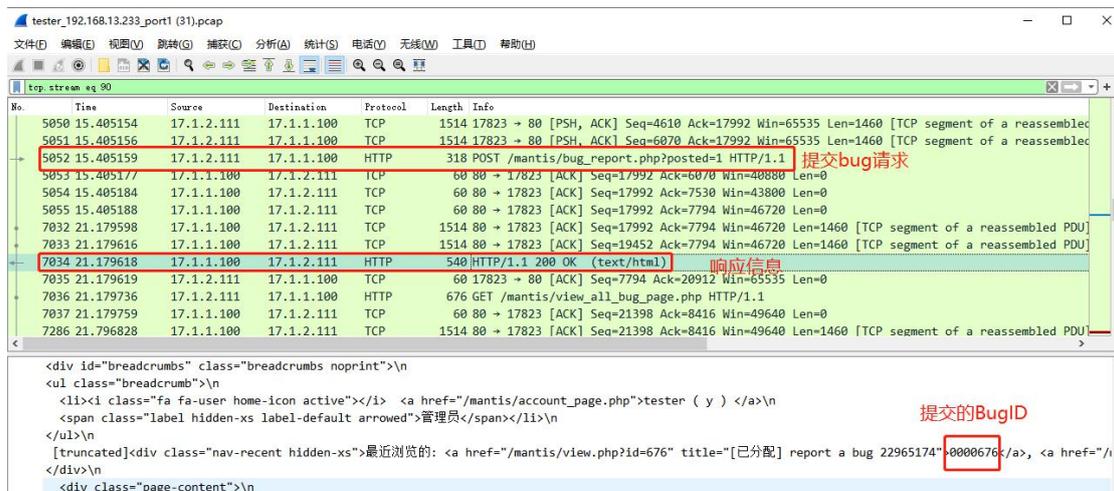
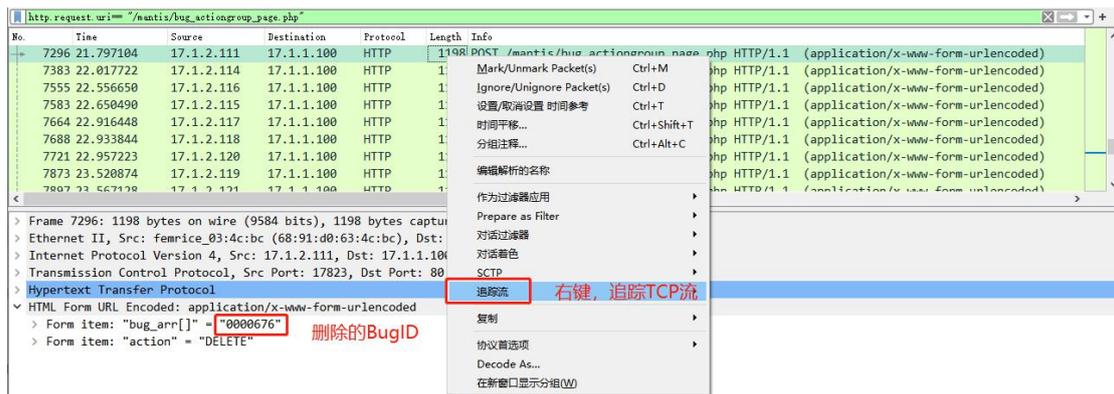
- Form item: "summary" = "BugSummary"

The capture also shows a detailed view of the packet structure, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol.



3) 按 BugID 删除 Bug

在过滤器输入 `http.request.uri== "/mantis/bug_actiongroup_page.php"`, 过滤出提交 Bug 页面, 选中查看报文信息中删除的 BugID 为 9332。跟踪 TCP 流, 找到提交 bug 的回应信息, 里边的 BugID 为 676, 删除时的 BugID 就是变量 ReportID 提取出来的 BugID。



3. 事务列表的其他功能

Supernova 测试仪还有一些附加的功能，例如文本替换、批量删除功能，禁用启用请求功能，和业务组功能。

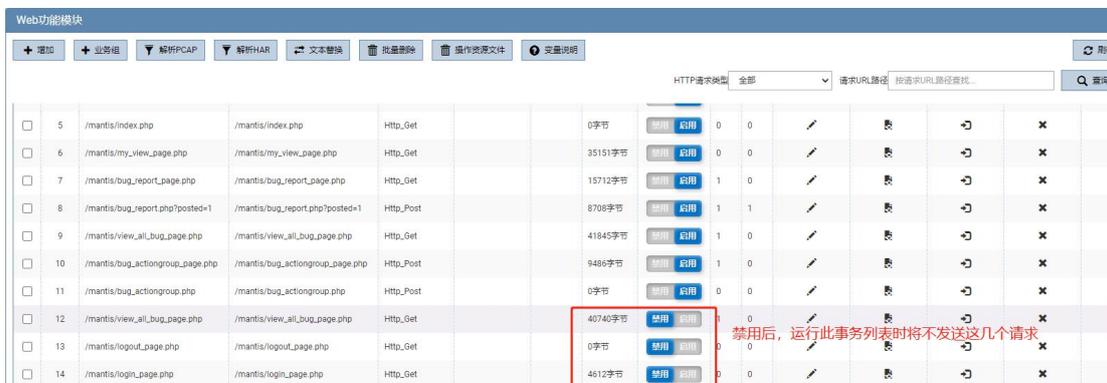
文本替换功能可以实现在“发送页面路径和文件名”、“服务器 IP 地址或域名”、“客户端请求头部/数据”、“服务器响应头部/数据”批量搜索替换字符串。



批量删除功能可以实现一次删除多个请求，示范如下



禁用启用请求功能，可以自定义事务列表中某些请求是否发送。



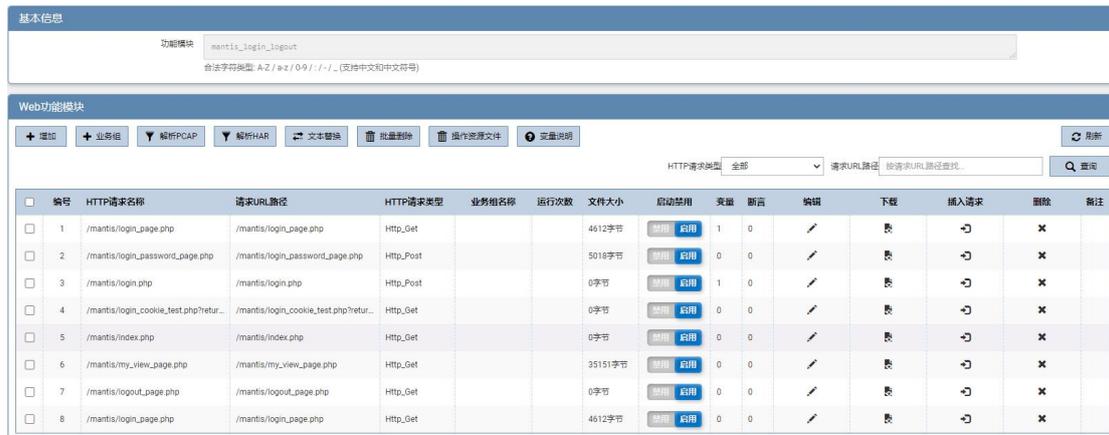
下面介绍业务组的功能。

3.1 业务组功能

业务组功能可以将该事务列表中某段编号的请求（例如 2-3 或者 2-2）组成一个业务组，可编辑设定其业务组运行次数，实现运行该事务列表时，某段编号的请求组成一个业务组循环多次执行。下面简单创建个事务列表并配置业务组功能。

3.5.1 创建 HTTP 事务列表

1) 参考 2.1 章节，创建一个新的 Web 功能模块，配置一个事务列表，业务逻辑为登录 mantis 后再退出，配置如下：



2) 点击“业务组”，进行业务中编辑配置，设置事务对象中，第 1-6 项请求为一个业务组，循环运行 2 次，配置完成后点击确定保存。



3) 点击确定保存后，事务组编辑完成，页面显示如下：

编号	HTTP请求名称	请求URL路径	HTTP请求类型	业务组名称	运行次数	文件大小	启动禁用	变量	断言	编辑	下载	插入请求	删除	备注
1	/mantis/login_page.php	/mantis/login_page.php	Http_Get	login	2	4612字节	禁用 启用	1	0		⬇️	➡️	✖️	
2	/mantis/login_password_page.p...	/mantis/login_password_page.p...	Http_Post	login	2	5018字节	禁用 启用	0	0		⬇️	➡️	✖️	
3	/mantis/login.php	/mantis/login.php	Http_Post	login	2	0字节	禁用 启用	1	0		⬇️	➡️	✖️	
4	/mantis/login_cookie_test.php?r...	/mantis/login_cookie_test.php?r...	Http_Get	login	2	0字节	禁用 启用	0	0		⬇️	➡️	✖️	
5	/mantis/index.php	/mantis/index.php	Http_Get	login	2	0字节	禁用 启用	0	0		⬇️	➡️	✖️	
6	/mantis/my_view_page.php	/mantis/my_view_page.php	Http_Get	login	2	35151字节	禁用 启用	0	0		⬇️	➡️	✖️	
7	/mantis/logout_page.php	/mantis/logout_page.php	Http_Get			0字节	禁用 启用	0	0		⬇️	➡️	✖️	
8	/mantis/login_page.php	/mantis/login_page.php	Http_Get			4612字节	禁用 启用	0	0		⬇️	➡️	✖️	

3.5.2 创建 HTTP 事务用例并运行

创建一个 HTTP 事务测试用例，引用刚刚创建的 HTTP 事务列表，并修改配置参数，比如端口的 IP 地址以及虚拟用户的数量。下面简单介绍步骤，详细的操作请参考 2.2 章节。

1) 依次点击，用例 -> 应用服务测试 -> HTTP -> 应用服务业务测试 -> 增加，就会弹出增加 HTTP 事务用例的对话框。



2) 输入 HTTP 事务用例的名称，设置参数信息并在动作中选择刚刚配置好的 HTTP 事务列表，在网络选项下设置好报文捕获，运行后通过抓包来验证插入其他事务的功能。为方便查看业务组功能效果，可将虚拟用户数量设置为 1。



3) HTTP 事务列表和测试用例配置完成之后，点击运行启动 HTTP 应用服务业务测试用例。



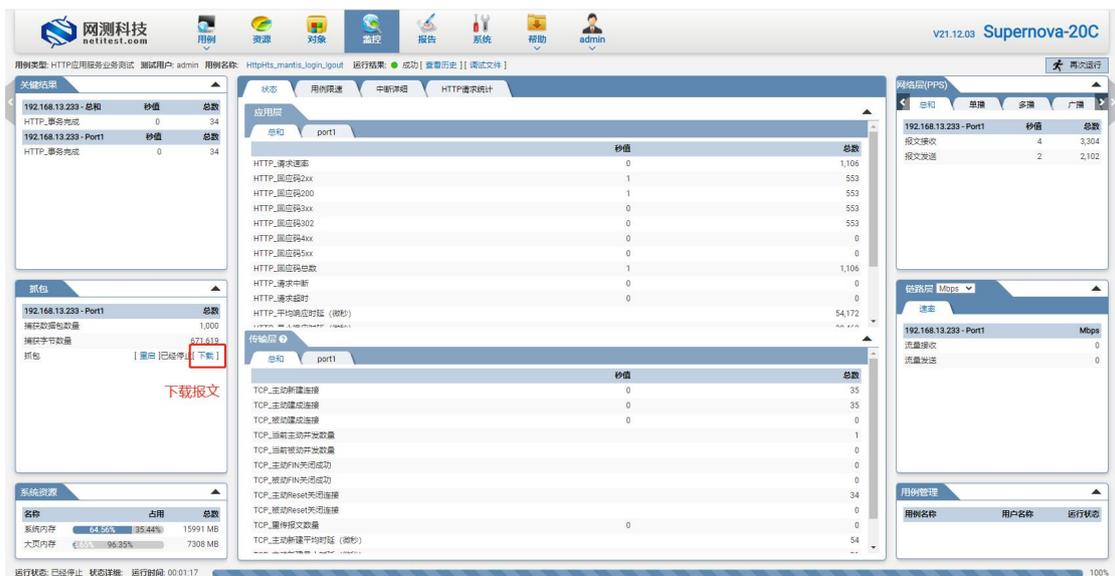
在运行页面 HTTP 请求统计，可以看出前 6 个请求运行 2 次后，才执行后边的两次请求。

事务名称	运行次数	运行速率	成功次数	成功速率	出错次数	平均模块完成时间(ms)	最小模块完成时间(ms)	最大模块完成时间(ms)
mantis_login_logout	10	1.4/sec	9	1.3/sec	0	722	694	804

事务名称	请求路径	HTTP请求次数	HTTP请求速率	HTTP响应次数	HTTP响应速率	断言失败次数	HTTP出错次数	HTTP出错百分比	发送速率
mantis_login_logout	/mantis/login_page.php	20	2.9/sec	20	2.9/sec	0	0	0.00%	
mantis_login_logout	/mantis/login_password_page.php	20	2.9/sec	20	2.9/sec	0	0	0.00%	
mantis_login_logout	/mantis/login.php	20	2.9/sec	20	2.9/sec	0	0	0.00%	
mantis_login_logout	/mantis/login_cookie_test.php?ret...	20	2.9/sec	19	2.7/sec	0	0	0.00%	
mantis_login_logout	/mantis/index.php	19	2.7/sec	19	2.7/sec	0	0	0.00%	
mantis_login_logout	/mantis/my_view_page.php	19	2.7/sec	19	2.7/sec	0	0	0.00%	
mantis_login_logout	/mantis/logout_page.php	9	1.3/sec	9	1.3/sec	0	0	0.00%	
mantis_login_logout	/mantis/login_page.php	9	1.3/sec	9	1.3/sec	0	0	0.00%	

3.5.3 抓包查看报文

由于在创建事务用例，配置网络时，我们设置了报文捕获。所以运行过程中或运行结束后，在监测页面可以下载 pcap 包，使用 wireshark 打开。



查看抓包内容，可以看出先请求了对象名称为 HttpObject_login_out 的事务，后请求了对象名称为 HttpObject 的事务，说明设置的插入其他事务功能起效。

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000169	17.1.2.2	17.1.1.100	HTTP	505	GET /mantis/login_page.php HTTP/1.1
6	0.002570	17.1.1.100	17.1.2.2	HTTP	790	HTTP/1.1 200 OK (text/html)
8	0.002676	17.1.2.2	17.1.1.100	HTTP	753	POST /mantis/login_password_page.php HTTP/1.1 (application/x-www-form-urlencoded)
11	0.124278	17.1.1.100	17.1.2.2	HTTP	762	HTTP/1.1 200 OK (text/html)
13	0.124334	17.1.2.2	17.1.1.100	HTTP	782	POST /mantis/login.php HTTP/1.1 (application/x-www-form-urlencoded)
15	0.215251	17.1.1.100	17.1.2.2	HTTP	955	HTTP/1.1 302 Found
16	0.215302	17.1.2.2	17.1.1.100	HTTP	756	GET /mantis/login_cookie_test.php?return-index.php HTTP/1.1
18	0.286211	17.1.1.100	17.1.2.2	HTTP	754	HTTP/1.1 302 Found
19	0.286243	17.1.2.2	17.1.1.100	HTTP	727	GET /mantis/index.php HTTP/1.1
21	0.324974	17.1.1.100	17.1.2.2	HTTP	761	HTTP/1.1 302 Found
22	0.325007	17.1.2.2	17.1.1.100	HTTP	734	GET /mantis/my_view_page.php HTTP/1.1
27	0.397656	17.1.1.100	17.1.2.2	HTTP	1463	HTTP/1.1 200 OK (text/html)
30	0.397804	17.1.2.2	17.1.1.100	HTTP	505	GET /mantis/login_page.php HTTP/1.1
33	0.439576	17.1.1.100	17.1.2.2	HTTP	790	HTTP/1.1 200 OK (text/html)
35	0.439622	17.1.2.2	17.1.1.100	HTTP	753	POST /mantis/login_password_page.php HTTP/1.1 (application/x-www-form-urlencoded)
38	0.481273	17.1.1.100	17.1.2.2	HTTP	762	HTTP/1.1 200 OK (text/html)
40	0.481313	17.1.2.2	17.1.1.100	HTTP	782	POST /mantis/login.php HTTP/1.1 (application/x-www-form-urlencoded)
42	0.547042	17.1.1.100	17.1.2.2	HTTP	955	HTTP/1.1 302 Found
43	0.547072	17.1.2.2	17.1.1.100	HTTP	756	GET /mantis/login_cookie_test.php?return-index.php HTTP/1.1
45	0.590169	17.1.1.100	17.1.2.2	HTTP	754	HTTP/1.1 302 Found
46	0.590191	17.1.2.2	17.1.1.100	HTTP	727	GET /mantis/index.php HTTP/1.1
48	0.633234	17.1.1.100	17.1.2.2	HTTP	761	HTTP/1.1 302 Found
49	0.633256	17.1.2.2	17.1.1.100	HTTP	734	GET /mantis/my_view_page.php HTTP/1.1
54	0.714459	17.1.1.100	17.1.2.2	HTTP	1463	HTTP/1.1 200 OK (text/html)
57	0.714621	17.1.2.2	17.1.1.100	HTTP	733	GET /mantis/logout_page.php HTTP/1.1
59	0.757506	17.1.1.100	17.1.2.2	HTTP	1272	HTTP/1.1 302 Found
60	0.757577	17.1.2.2	17.1.1.100	HTTP	599	GET /mantis/login_page.php HTTP/1.1
63	0.802905	17.1.1.100	17.1.2.2	HTTP	790	HTTP/1.1 200 OK (text/html)

前6个请求循环运行两次后
执行最后两个请求