FINDING VULNERABILITIES IN OPEN SOURCE IP PBX VOIP SOFTWARE

PIERRE JOURDAN

WHOAMI

- Pierre Jourdan
- Technical Support Manager @ 3CX
- I'm a geek & Passionate about Cyber Security!
- Multiple certifications in Cyber Security and Data Privacy
  MSc Cyber-Sec, ISACA CISM & CISA, IAPP CIPP/E, CIPT, CIPM, FIP, PECB CDPO
- White hat hacker…

DISCLAIMER

https://icon-library.net

- *Opinions expressed are solely my own and do not express the views or opinions of my employer.*
- The presentation is intended for educational purposes only.

The idea

- Open source software code is **easy to audit** as disclosed
- They have **many contributors**, changing over time, with **various security awareness**
- Q&A? SDLC?
- They have **limited** resources / budget
- I know about VoIP and Cyber-Security is a passion…

So…

- Let's see if we can **find vulnerabilities** in some well-known VoIP software ☺

- Also no budget here, doing all with free tools on my spare time for past 3 months
- Still a work-in-progress, spent around **100 Hrs** on it
- **Responsible disclosure** of the findings
- Lessons learned and **mitigation**

GOOGLING

- Who are the **main open source VoIP** sofware nowadays?
- Found 3 main actors: **Asterisk, FreeSwitch, Kamailio**
- **Many frontends/distro** are built around these core engines
- **Aging** tech, > 10y old for some
- Core in C/C++, frontends in PHP/SQL/HTML/JS

What's under the hood

- Run in Linux, various **distro** (CentOS, Debian)
- Uses various **web** servers (Apache, Nginx)
- Uses various **SQL** servers (PostgreSQL, MySQL)
- Administration panel in **PHP**/SQL/HTML/JS
- Various **SIP** engines for the communication with handsets/softphones/trunks
  - Asterisk uses PJ_SIP or CHAN_SIP (older)
  - Freeswitch uses Sofia
  - Kamailio is based on Openser
- Source code in GitHub, **ISO** ready to use and download

TEST ENVIRONMENT

- Installing ISO's in **VMware Player**
- Some quick tests and **reconnaissance**
- Downloaded latest **sources** from GitHub
- Keep track of the findings with Mantis (**bug tracker**), installed quickly in an EasyPHP (web-server kit for windows)
- Get started?

Where to start?

- Focusing on FusionPBX first
- Focusing on its **administration panel**

- Handy tools: Notepad++ and SearchMyFiles

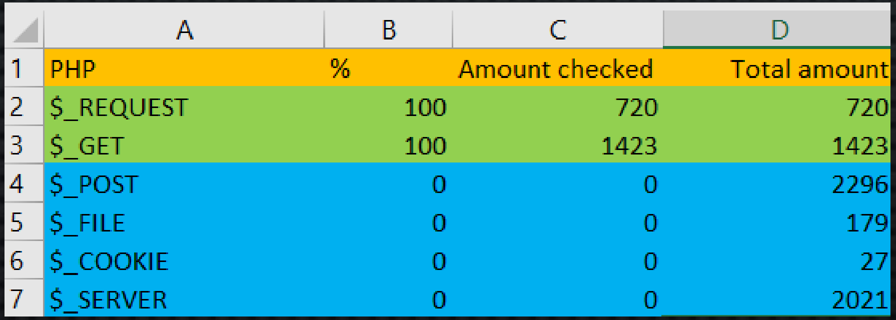
METHOD 1: AUDIT USER INPUTS

- Searching **user inputs** variables in PHP

- Look for all occurrences of:
  - $_REQUEST[‘xxx’] (HTTP requests arguments)
  - $_GET[‘xxx’] (variables passed in URLs)
  - $_POST[‘xxx’] (variables passed in forms)
  - $_COOKIE[‘xxx’] (variables passed in cookies)
  - $_SERVER[‘xxx’] (some variables passed by user browser, such as user-agent, referer)
  - $_FILE[‘xxx’] (File uploads variables such as filename)

Method 1: Audit User Inputs

- How? Search for « $_ » in SearchMyFiles, going through the source repo recursively
- Open all files found in notepad++
- Search again in all opened documents, go one by one
- Make a spreadsheet with total occurrences and amount reviewed, with a percent representing the code coverage, keep track of progress



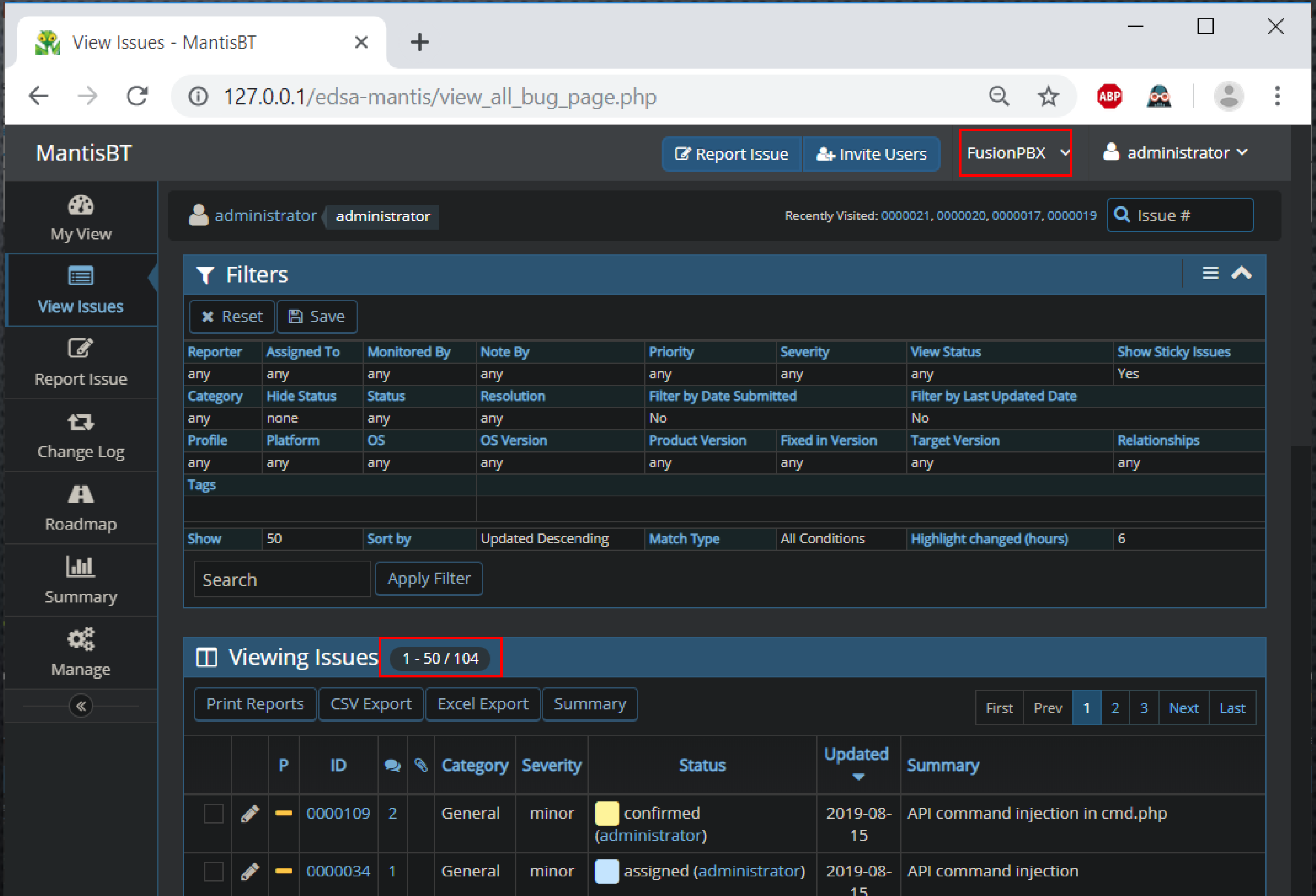
	A	B	C	D
1	PHP	%	Amount checked	Total amount
2	$_REQUEST	100	720	720
3	$_GET	100	1423	1423
4	$_POST	0	0	2296
5	$_FILE	0	0	179
6	$_COOKIE	0	0	27
7	$_SERVER	0	0	2021

METHOD 1: AUDIT USER INPUTS

- What type of vulnerabilities are we looking for?
  - Cross-site scripting (**XSS**), with variables reflected directly in html
  - SQL injections (**SQLi**), with variables inserted directly in SQL queries
  - Others **injections**, e.g Path traversals
- Any luck?
- YES!!!

FusionPBX Findings in Brief

- XSS: 33
- SQLi: 55
- Others*: 15
  - I stopped at **100** vulnerabilities got tired!! There are more ☺
- Examples?



administrator administrator

Recently Visited: 0000021, 0000020, 0000017, 0000019

Issue #

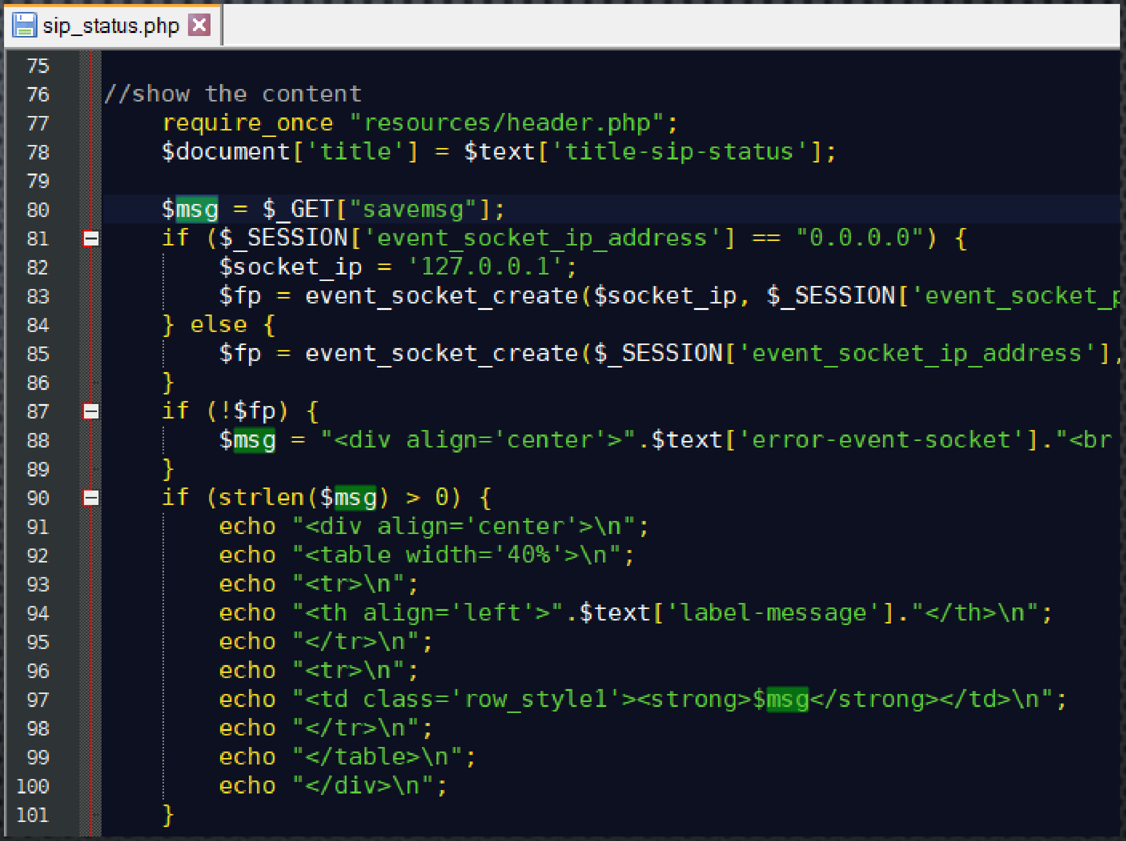
Reporter	Assigned To	Monitored By	Note By	Priority	Severity	View Status	Show Sticky Issues
any	any	any	any	any	any	any	Yes
Category	Hide Status	Status	Resolution	Filter by Date Submitted		Filter by Last Updated Date	
any	none	any	any	No		No	
Profile	Platform	OS	OS Version	Product Version	Fixed in Version	Target Version	Relationships
any	any	any	any	any	any	any	any
Tags							
Show	50	Sort by	Updated Descending	Match Type	All Conditions	Highlight changed (hours)	6

Viewing Issues 1 - 50 / 104

		P	ID	💬	📎	Category	Severity	Status	Updated	Summary
□	✎	—	0000109	2		General	minor	confirmed (administrator)	2019-08-15	API command injection in cmd.php
□	✎	—	0000034	1		General	minor	assigned (administrator)	2019-08-15	API command injection

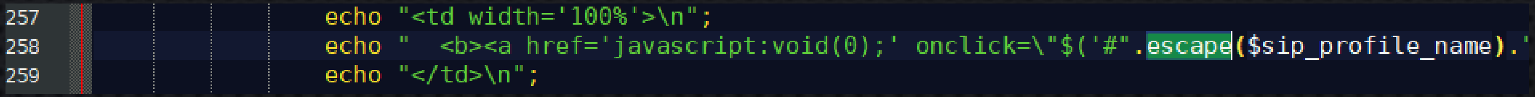
XSS – EXAMPLE

- An example out of many, of a **GET** variable taken as-is and **reflected** in HTML code generated by one of the pages



```
//show the content
    require_once "resources/header.php";
    $document['title'] = $text['title-sip-status'];

    $msg = $_GET["savemsg"];
    if ($_SESSION['event_socket_ip_address'] == "0.0.0.0") {
        $socket_ip = '127.0.0.1';
        $fp = event_socket_create($socket_ip, $_SESSION['event_socket_p
    } else {
        $fp = event_socket_create($_SESSION['event_socket_ip_address'],
    }
    if (!$fp) {
        $msg = "<div align='center'>".$text['error-event-socket']."<br
    }
    if (strlen($msg) > 0) {
        echo "<div align='center'>\n";
        echo "<table width='40%'>\n";
        echo "<tr>\n";
        echo "<th align='left'>".$text['label-message']."</th>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td class='row_style1'><strong>$msg</strong></td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "</div>\n";
    }
```

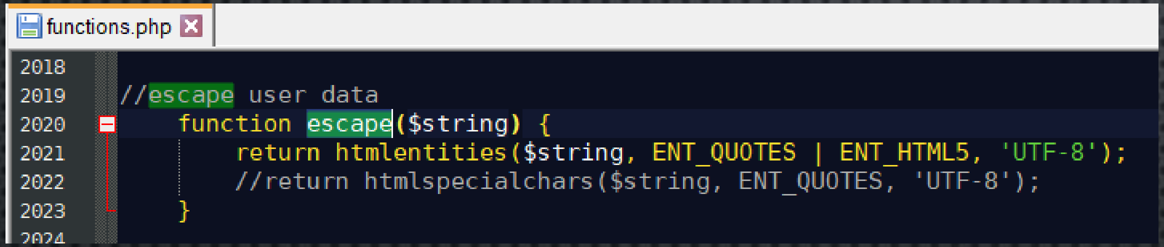
- Whilst on others variables in same page and others, are normally put under an **escape()** function for clean-up/**sanitization**



```
                echo "<td width='100%'>\n";
                echo "  <b><a href='javascript:void(0);' onclick=\"$('#".escape($sip_profile_name).'
                echo "</td>\n";
```

XSS – ESCAPE() FUNCTION

- The escape() function is just passing args to the PHP function **htmlentities**()



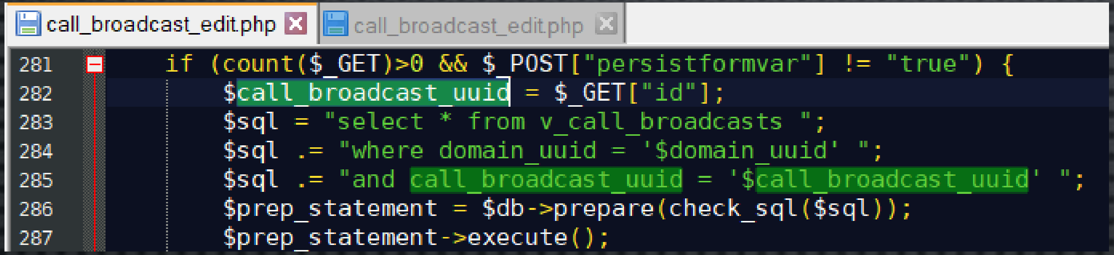
functions.php

```
//escape user data
    function escape($string) {
        return htmlentities($string, ENT_QUOTES | ENT_HTML5, 'UTF-8');
        //return htmlspecialchars($string, ENT_QUOTES, 'UTF-8');
    }
```

- **Alert(1)** becomes **alert&lpar;1&rpar;** once escaped

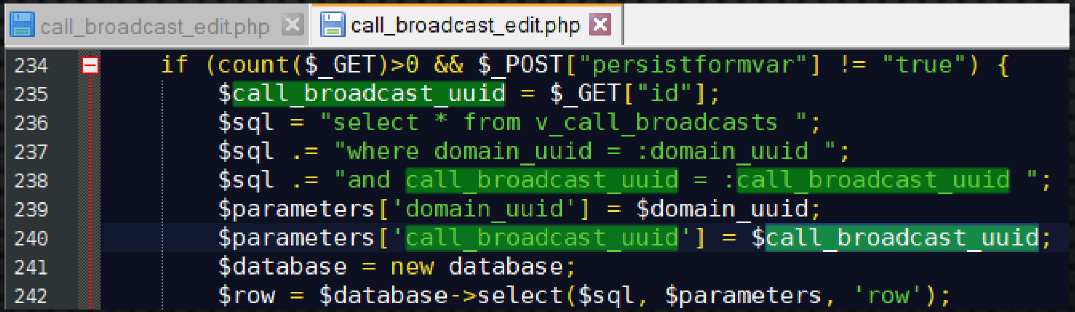
SQLI - EXAMPLE

- When first checked I found **many SQL injections** but until submit a month later they had fixed most, yet they **forgot this one**:



```php
	if (count($_GET)>0 && $_POST["persistformvar"] != "true") {
		$call_broadcast_uuid = $_GET["id"];
		$sql = "select * from v_call_broadcasts ";
		$sql .= "where domain_uuid = '$domain_uuid' ";
		$sql .= "and call_broadcast_uuid = '$call_broadcast_uuid' ";
		$prep_statement = $db->prepare(check_sql($sql));
		$prep_statement->execute();
```

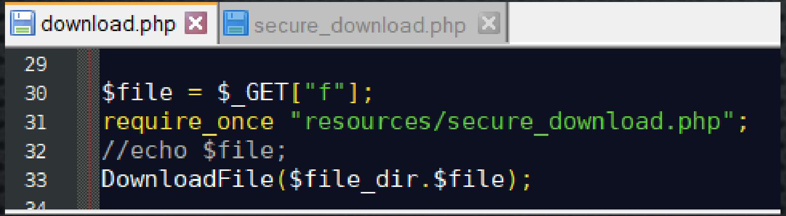
- Here we see a GET variable taken as-is in an SQL query syntax
- Once reported the fix was to use **parameterization** (as everywhere else)



```php
	if (count($_GET)>0 && $_POST["persistformvar"] != "true") {
		$call_broadcast_uuid = $_GET["id"];
		$sql = "select * from v_call_broadcasts ";
		$sql .= "where domain_uuid = :domain_uuid ";
		$sql .= "and call_broadcast_uuid = :call_broadcast_uuid ";
		$parameters['domain_uuid'] = $domain_uuid;
		$parameters['call_broadcast_uuid'] = $call_broadcast_uuid;
		$database = new database;
		$row = $database->select($sql, $parameters, 'row');
```

PATH TRAVERSAL

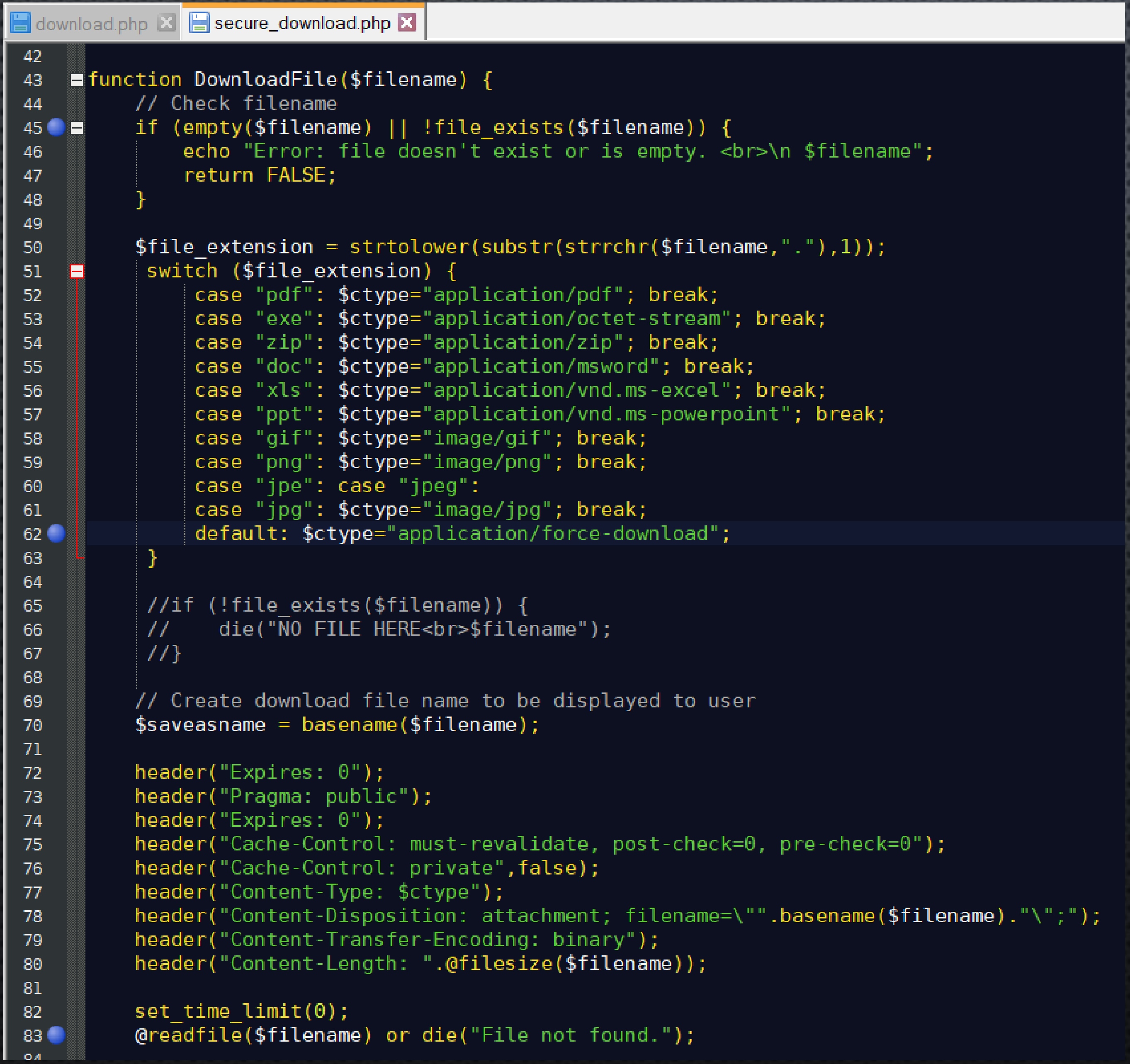
- FEW PLACES WHERE FILE OPERATIONS WERE POSSIBLE WITHOUT PROPER **SANITIZATION OF THE FILE OR FOLDER** ARGS
- ALLOWS TO TAMPER WITH ANY FILE OF THE SYSTEM
- Here we see a download page which can be called by any authenticated user passing GET arg to a subfunction:



```php

$file = $_GET["f"];
require_once "resources/secure_download.php";
//echo $file;
DownloadFile($file_dir.$file);
```

- Interestingly the $file_dir variable is never set…

PATH TRAVERSAL - SUBFUNCTION



```php

function DownloadFile($filename) {
    // Check filename
    if (empty($filename) || !file_exists($filename)) {
        echo "Error: file doesn't exist or is empty. <br>\n $filename";
        return FALSE;
    }

    $file_extension = strtolower(substr(strrchr($filename,"."),1));
     switch ($file_extension) {
         case "pdf": $ctype="application/pdf"; break;
         case "exe": $ctype="application/octet-stream"; break;
         case "zip": $ctype="application/zip"; break;
         case "doc": $ctype="application/msword"; break;
         case "xls": $ctype="application/vnd.ms-excel"; break;
         case "ppt": $ctype="application/vnd.ms-powerpoint"; break;
         case "gif": $ctype="image/gif"; break;
         case "png": $ctype="image/png"; break;
         case "jpe": case "jpeg":
         case "jpg": $ctype="image/jpg"; break;
         default: $ctype="application/force-download";
     }

     //if (!file_exists($filename)) {
     //    die("NO FILE HERE<br>$filename");
     //}

    // Create download file name to be displayed to user
    $saveasname = basename($filename);

    header("Expires: 0");
    header("Pragma: public");
    header("Expires: 0");
    header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
    header("Cache-Control: private",false);
    header("Content-Type: $ctype");
    header("Content-Disposition: attachment; filename=\"".basename($filename)."\";");
    header("Content-Transfer-Encoding: binary");
    header("Content-Length: ".@filesize($filename));

    set_time_limit(0);
    @readfile($filename) or die("File not found.");
```

- Check if file exists only

- CHECK FILE EXTENSION, FOR HTTP HEADER PURPOSE ONLY, IF NO MATCH, FORCE DOWNLOAD!

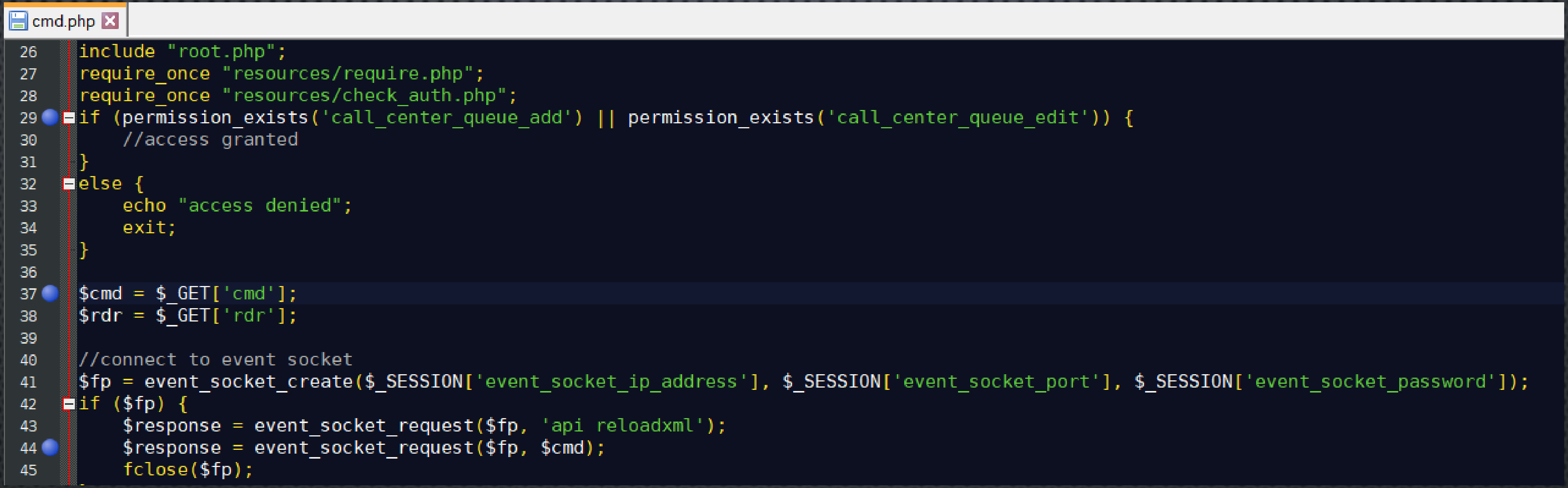
- Read file and output without path control

- E.G: https://xxx/resources/download.php?f=/etc/passwd

OTHERS

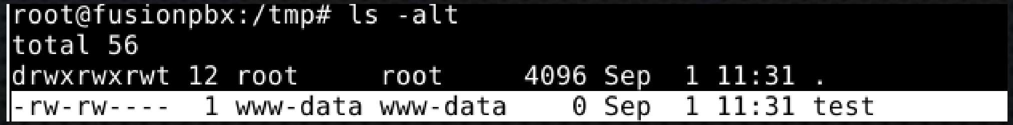
- Sofia (Freeswitch SIP engine) provides a **LUA API** in which instructions can be sent to the core
- Commands passed through a socket accessible to **localhost**
- Most of these commands are doing actions in PBX only
- Interestingly, not much documented is a call allowing to run **system commands**
- So, if not properly implemented someone could compromise the machine completely from web files

API INJECTION EXPLAINED



```php
include "root.php";
require_once "resources/require.php";
require_once "resources/check_auth.php";
if (permission_exists('call_center_queue_add') || permission_exists('call_center_queue_edit')) {
    //access granted
}
else {
    echo "access denied";
    exit;
}

$cmd = $_GET['cmd'];
$rdr = $_GET['rdr'];

//connect to event socket
$fp = event_socket_create($_SESSION['event_socket_ip_address'], $_SESSION['event_socket_port'], $_SESSION['event_socket_password']);
if ($fp) {
    $response = event_socket_request($fp, 'api reloadxml');
    $response = event_socket_request($fp, $cmd);
    fclose($fp);
```

- We see first that the page is testing if a specific permission exists (lower than admin priviledge) = potential priviledge escalation
- Then a GET arg is taken as-is (again) and sent to the socket
- https://XXX/app/call_centers/cmd.php?rdr=false &cmd=**api**%20**system**%20**touch**%20**/tmp/test**
- Results in a file created in /tmp folder, as user **www-data**



```
root@fusionpbx:/tmp# ls -alt
total 56
drwxrwxrwt 12 root     root     4096 Sep  1 11:31 .
-rw-rw----  1 www-data www-data    0 Sep  1 11:31 test
```

METHOD 2: CHECK DEPENDANCIES

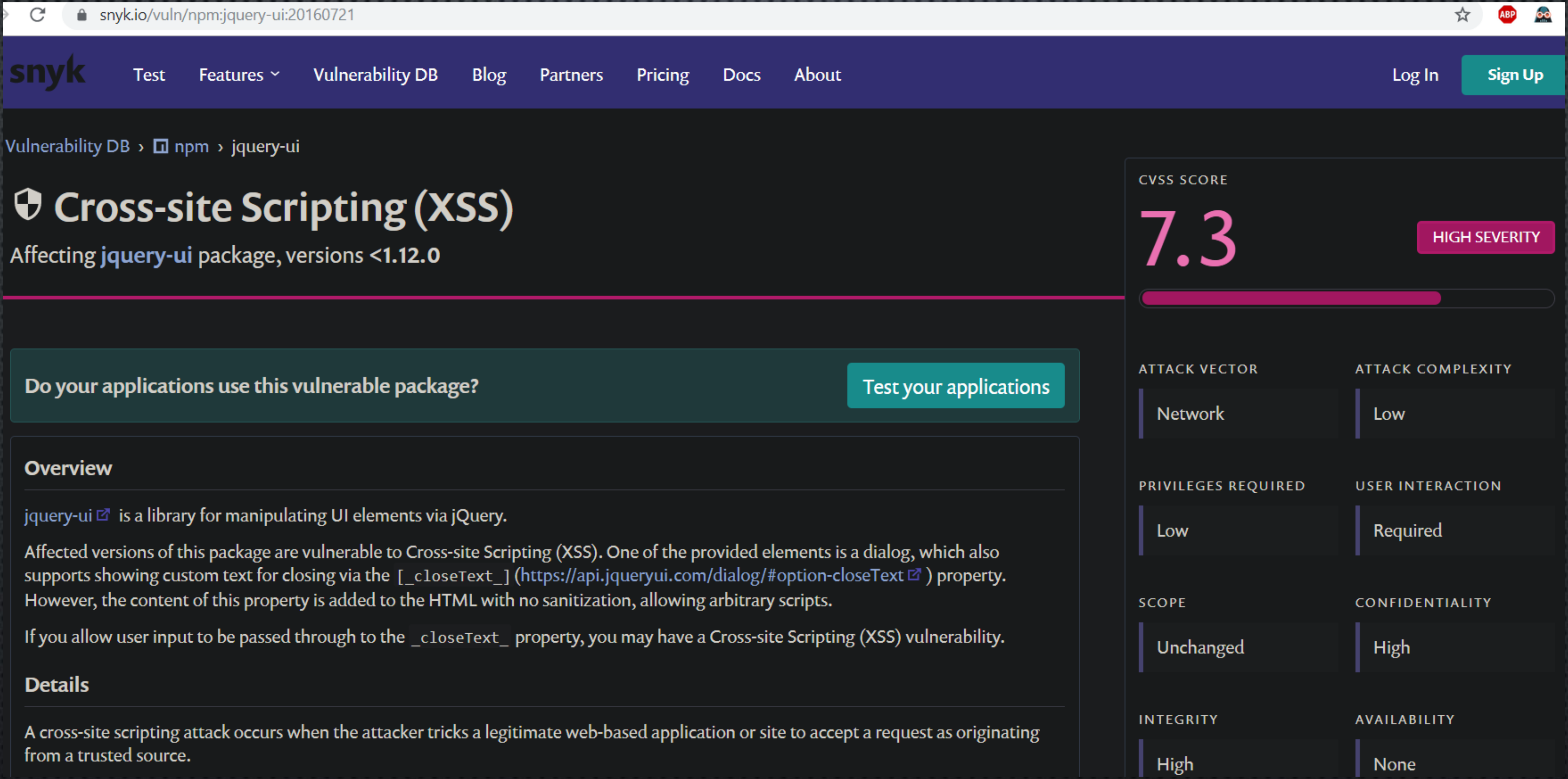
- LIST ALL **JAVASCRIPT** FILES FROM SOURCES
- CHECK THE COMMENTS/HEADER LOOKING FOR APP **NAME AND VERSION**
- SEARCH FOR THOSE ON **SNYK.IO**

- WHAT ARE WE LOOKING FOR?
  - OLD DEPENDANCIES, FORGOTTEN AND **OUT-OF-DATE**
  - KNOWN ISSUES IN VULNERABLE JAVASCRIPT PACKAGES

- ANY LUCK?
- YES!!!

METHOD 2: CHECK DEPENDANCIES

- 5 REPORTED VULNERABILITIES IN JS DEPENDANCIES:
  - MULTIPLE XSS IN JQUERY 1.8.3 AND 1.11.1
  - MULTIPLE XSS IN JQUERY UI 1.9.2
  - MULTIPLE XSS IN BOOTSTRAP 3.3.6
  - JS PROTOTYPE POLLUTION IN JQUERY 1.8.3



Vulnerability DB › npm › jquery-ui

Cross-site Scripting (XSS)

Affecting **jquery-ui** package, versions **<1.12.0**

Do your applications use this vulnerable package?

Test your applications

Overview

jquery-ui is a library for manipulating UI elements via jQuery.

Affected versions of this package are vulnerable to Cross-site Scripting (XSS). One of the provided elements is a dialog, which also supports showing custom text for closing via the `[_closeText_]` (https://api.jqueryui.com/dialog/#option-closeText) property. However, the content of this property is added to the HTML with no sanitization, allowing arbitrary scripts.

If you allow user input to be passed through to the `_closeText_` property, you may have a Cross-site Scripting (XSS) vulnerability.

Details

A cross-site scripting attack occurs when the attacker tricks a legitimate web-based application or site to accept a request as originating from a trusted source.

CVSS SCORE

7.3

HIGH SEVERITY

ATTACK VECTOR	ATTACK COMPLEXITY
Network	Low
PRIVILEGES REQUIRED	**USER INTERACTION**
Low	Required
SCOPE	**CONFIDENTIALITY**
Unchanged	High
INTEGRITY	**AVAILABILITY**
High	None

METHOD 3: ANYTHING UNAUTHENTICATED?

- A VALUABLE ATTACK VECTOR IS A PAGE THAT CAN BE REQUESTED FROM REMOTE WHILST **UNAUTHENTICATED**.
- AS PBX DISTROS HAVE HUNDREDS/THOUSANDS OF PAGES, WE NEED AN AUTOMATED **CRAWLER**
- SO I MADE A QUICK PHP SCRIPT **LISTING ALL PHP FILES** OF THE DISTRO FOLDER, AND BROWSING THEM **RECURSIVELY** THROUGH A CURL FUNCTION ON MY WEBSERVER.
- UNFORTUNATELY, AFTER REVIEW, THERE WERE **NO FINDINGS** WITH THIS METHOD.

WHAT'S NEXT?

- Rechecked the findings on latest version (a month had passed)
- Tested them in my browser, play with URLs and **INSPECT** source with **dev tools**.
- For POST args or more complex tampering, used **BURP** community
- Reported the ones still current in vendor bug tracker (40 out of 100)
- Most got fixed fast, dev thankful

- Informed them of willing to fill CVE about those and disclosure in 1 month
- Requested 35 CVE numbers on https://cve.mitre.org/cve/request_id.html
  Waiting validation… And a week ago got confirm with CVE reserved nums!

RESERVED CVES

- CVE-2019-16964
- CVE-2019-16965
- CVE-2019-16966
- CVE-2019-16967
- CVE-2019-16968
- CVE-2019-16969
- CVE-2019-16970
- CVE-2019-16971
- CVE-2019-16972
- CVE-2019-16973
- CVE-2019-16974
- CVE-2019-16975
- CVE-2019-16976
- CVE-2019-16977
- CVE-2019-16978
- CVE-2019-16979
- CVE-2019-16980
- CVE-2019-16981
- CVE-2019-16982
- CVE-2019-16983
- CVE-2019-16984
- CVE-2019-16985
- CVE-2019-16986
- CVE-2019-16987
- CVE-2019-16988
- CVE-2019-16989
- CVE-2019-16990
- CVE-2019-16991

+ SOME MORE COMING ☺

FREEPBX

FreePBX®
let freedom ring™

- Same techniques were applied on FreePBX 14
- Found **10 XSS**, reported, and filled CVEs requests, also waiting
- Found vulnerable SQL queries, but turned out to be not exploitable
- didn't go deep…

SOME DIFFICULTIES

- Some code look **vulnerable but in practice aren't exploitable,** files aren't reachable directly as in FusionPBX, they pass through some central endpoints / Ajax handlers. Those are also escaping/encoding args.

- Chrome was blocking some of my XSS attempts, had to disable **XSS Auditor**
  => Launch "chrome.exe" **--disable-xss-auditor**

- Ran method 3 to locate files accessible without authentication, got banned!
  => had to **disable FAIL2BAN** on the VM.

Automating Audit

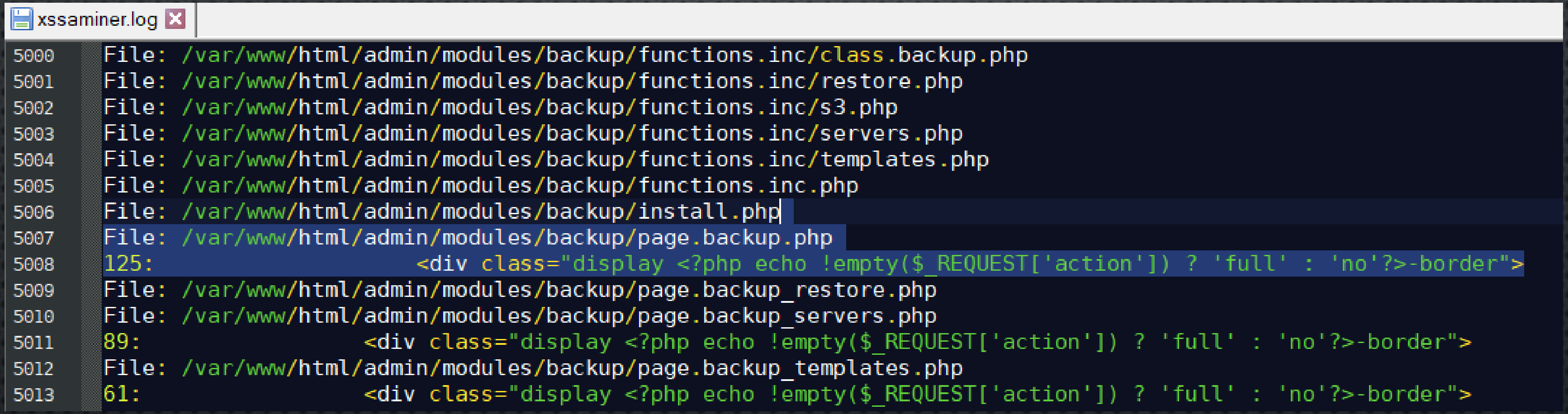
- Automate the code auditing part, there are good commercial solutions for that but expensive and not many free open-source alternatives…

- Found **XSSAMINER**, small shell script looking for XSS patterns in PHP code
  => lots of false-positives, but **found 3 valid XSS** out of it.

- Another well-known one is **SonarQube**, has a community edition and a commercial
  => Tried installing latest community in Ubuntu and Windows but **failed**…
  => **Lack of documentation** (on purpose?)
  => Next, will try **DOCKER** pre-installed/pre-configured

XSSAMINER LOGS

- Lots of false positives and things to cleanup but some findings are worthy



```
File: /var/www/html/admin/modules/backup/functions.inc/class.backup.php
File: /var/www/html/admin/modules/backup/functions.inc/restore.php
File: /var/www/html/admin/modules/backup/functions.inc/s3.php
File: /var/www/html/admin/modules/backup/functions.inc/servers.php
File: /var/www/html/admin/modules/backup/functions.inc/templates.php
File: /var/www/html/admin/modules/backup/functions.inc.php
File: /var/www/html/admin/modules/backup/install.php
File: /var/www/html/admin/modules/backup/page.backup.php
125:                    <div class="display <?php echo !empty($_REQUEST['action']) ? 'full' : 'no'?>-border">
File: /var/www/html/admin/modules/backup/page.backup_restore.php
File: /var/www/html/admin/modules/backup/page.backup_servers.php
89:                 <div class="display <?php echo !empty($_REQUEST['action']) ? 'full' : 'no'?>-border">
File: /var/www/html/admin/modules/backup/page.backup_templates.php
61:                 <div class="display <?php echo !empty($_REQUEST['action']) ? 'full' : 'no'?>-border">
```

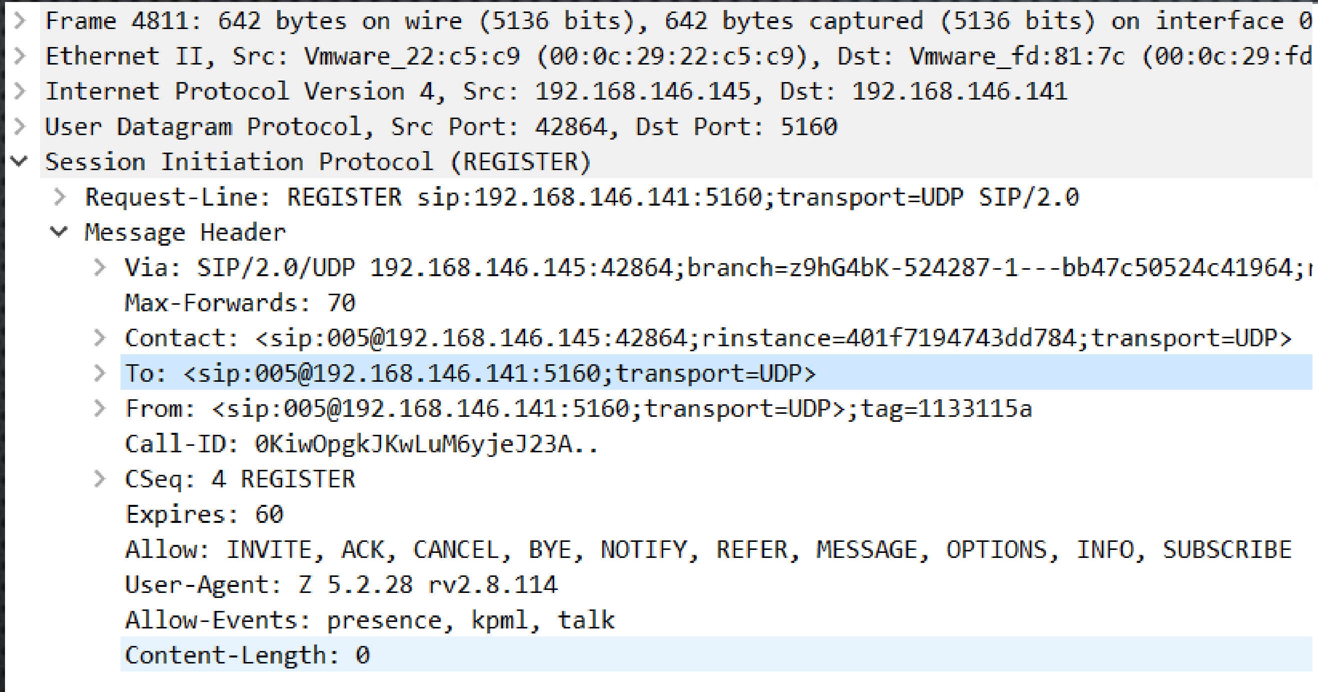
SIP BACKEND

- Focusing next on the **backend** (Asterisk & Freeswitch)
- **SIP** (Session Initiation Protocol) is a signalling protocol, widely used in VoIP
- Used for e.g to:
  - **Authenticate** phone endpoints (**REGISTER**)
  - Place **calls** (**INVITE**)
  - And many more..
- PBXs listen on SIP port **5060** TCP/UDP by default

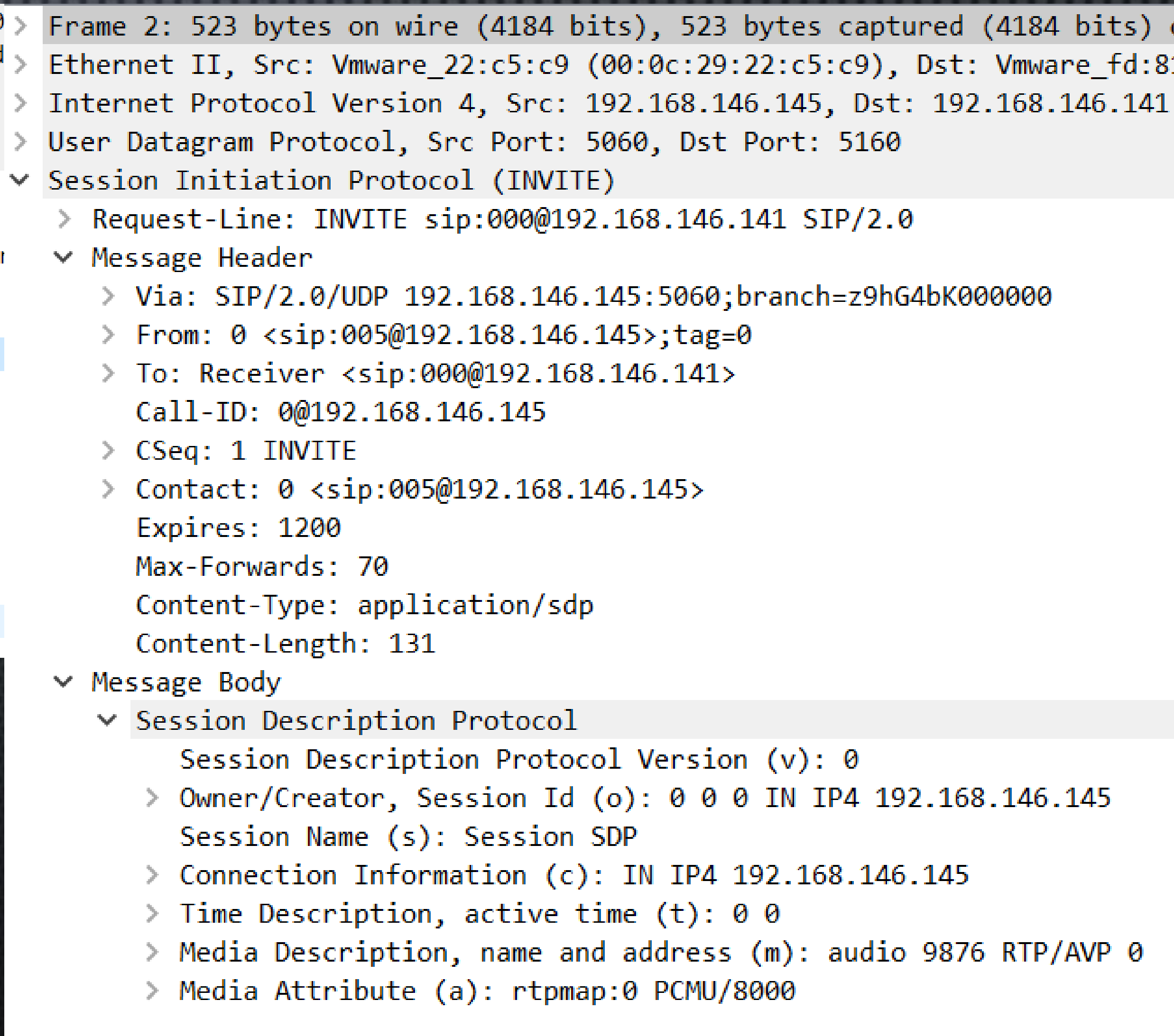
- What if we send invalid SIP messages? Any **Crashes/DoS** possible?

TYPICAL SIP FLOW

- Register as extension, INVITE from client (UAC), then answer from PBX (UAS):



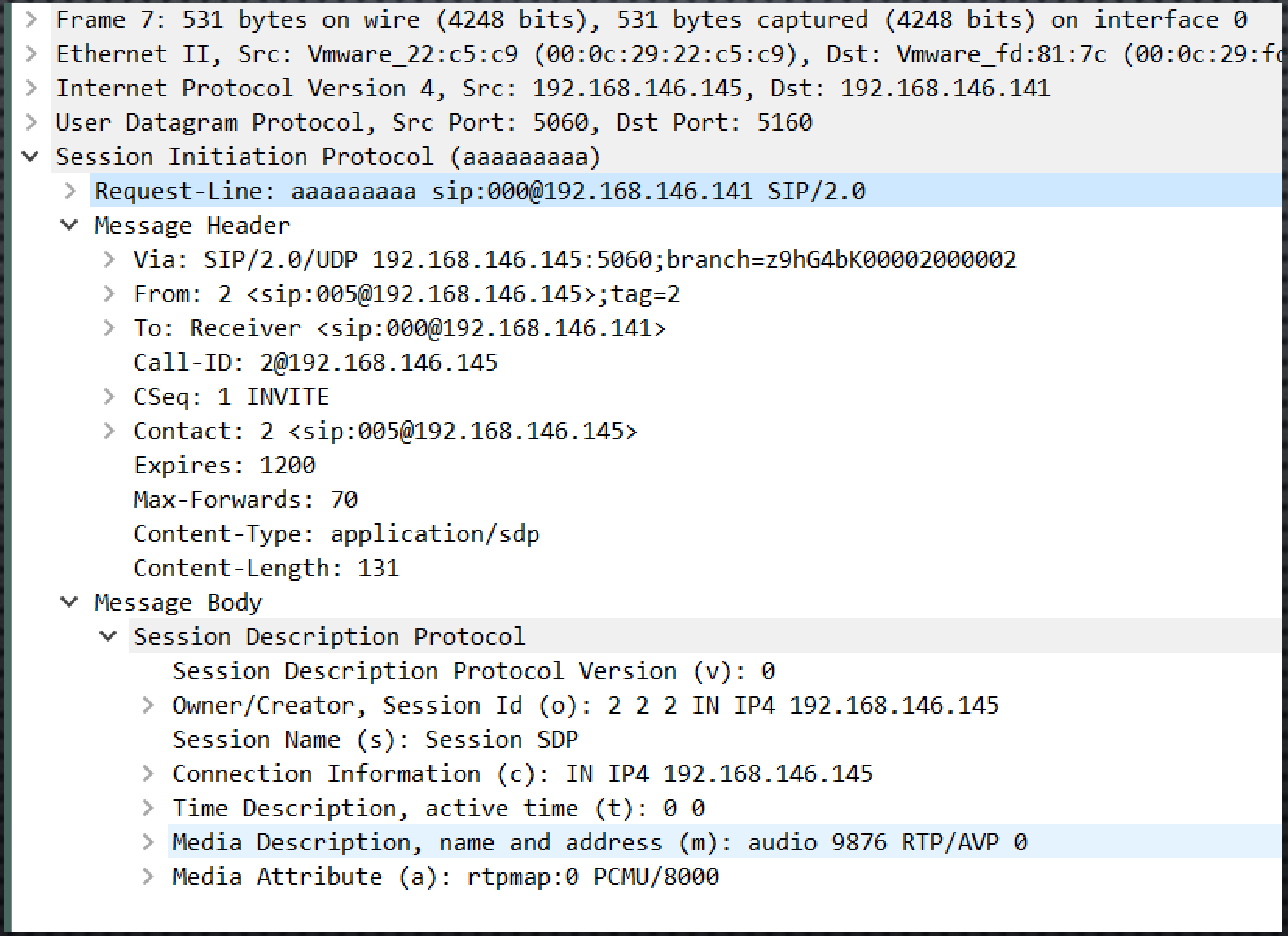
```
Frame 4811: 642 bytes on wire (5136 bits), 642 bytes captured (5136 bits) on interface 0
Ethernet II, Src: Vmware_22:c5:c9 (00:0c:29:22:c5:c9), Dst: Vmware_fd:81:7c (00:0c:29:fd
Internet Protocol Version 4, Src: 192.168.146.145, Dst: 192.168.146.141
User Datagram Protocol, Src Port: 42864, Dst Port: 5160
Session Initiation Protocol (REGISTER)
    Request-Line: REGISTER sip:192.168.146.141:5160;transport=UDP SIP/2.0
    Message Header
        Via: SIP/2.0/UDP 192.168.146.145:42864;branch=z9hG4bK-524287-1---bb47c50524c41964;
        Max-Forwards: 70
        Contact: <sip:005@192.168.146.145:42864;rinstance=401f7194743dd784;transport=UDP>
        To: <sip:005@192.168.146.141:5160;transport=UDP>
        From: <sip:005@192.168.146.141:5160;transport=UDP>;tag=1133115a
        Call-ID: 0KiwOpgkJKwLuM6yjeJ23A..
        CSeq: 4 REGISTER
        Expires: 60
        Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE, OPTIONS, INFO, SUBSCRIBE
        User-Agent: Z 5.2.28 rv2.8.114
        Allow-Events: presence, kpml, talk
        Content-Length: 0
```



```
Frame 2: 523 bytes on wire (4184 bits), 523 bytes captured (4184 bits)
Ethernet II, Src: Vmware_22:c5:c9 (00:0c:29:22:c5:c9), Dst: Vmware_fd:8
Internet Protocol Version 4, Src: 192.168.146.145, Dst: 192.168.146.141
User Datagram Protocol, Src Port: 5060, Dst Port: 5160
Session Initiation Protocol (INVITE)
    Request-Line: INVITE sip:000@192.168.146.141 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP 192.168.146.145:5060;branch=z9hG4bK000000
        From: 0 <sip:005@192.168.146.145>;tag=0
        To: Receiver <sip:000@192.168.146.141>
        Call-ID: 0@192.168.146.145
        CSeq: 1 INVITE
        Contact: 0 <sip:005@192.168.146.145>
        Expires: 1200
        Max-Forwards: 70
        Content-Type: application/sdp
        Content-Length: 131
    Message Body
        Session Description Protocol
            Session Description Protocol Version (v): 0
            Owner/Creator, Session Id (o): 0 0 0 IN IP4 192.168.146.145
            Session Name (s): Session SDP
            Connection Information (c): IN IP4 192.168.146.145
            Time Description, active time (t): 0 0
            Media Description, name and address (m): audio 9876 RTP/AVP 0
            Media Attribute (a): rtpmap:0 PCMU/8000
```

INSERTING ANOMALIES

- EXAMPLE ANOMALY: INSERT JUNK IN THE SIP METHOD



```
> Frame 7: 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface 0
> Ethernet II, Src: Vmware_22:c5:c9 (00:0c:29:22:c5:c9), Dst: Vmware_fd:81:7c (00:0c:29:f
> Internet Protocol Version 4, Src: 192.168.146.145, Dst: 192.168.146.141
> User Datagram Protocol, Src Port: 5060, Dst Port: 5160
v Session Initiation Protocol (aaaaaaaaa)
  > Request-Line: aaaaaaaaa sip:000@192.168.146.141 SIP/2.0
  v Message Header
    > Via: SIP/2.0/UDP 192.168.146.145:5060;branch=z9hG4bK00002000002
    > From: 2 <sip:005@192.168.146.145>;tag=2
    > To: Receiver <sip:000@192.168.146.141>
      Call-ID: 2@192.168.146.145
    > CSeq: 1 INVITE
    > Contact: 2 <sip:005@192.168.146.145>
      Expires: 1200
      Max-Forwards: 70
      Content-Type: application/sdp
      Content-Length: 131
  v Message Body
    v Session Description Protocol
        Session Description Protocol Version (v): 0
      > Owner/Creator, Session Id (o): 2 2 2 IN IP4 192.168.146.145
        Session Name (s): Session SDP
      > Connection Information (c): IN IP4 192.168.146.145
      > Time Description, active time (t): 0 0
      > Media Description, name and address (m): audio 9876 RTP/AVP 0
      > Media Attribute (a): rtpmap:0 PCMU/8000
```

- AS IT'S A COMPLEX PROTOCOL, AND THERE ARE MANY/INFINITE ANOMALIES, THOSE SHOULD BE TESTED AUTOMATICALLY THROUGH **FUZZING**, HERE FIRST TEST FROM PROTOS

OPEN SOURCE SIP FUZZERS

- Fuzzing with tools that are SIP-aware, to limit the anomalies « keyspace »

- **Protos** (16y old)
- **Voiper** (11y old)

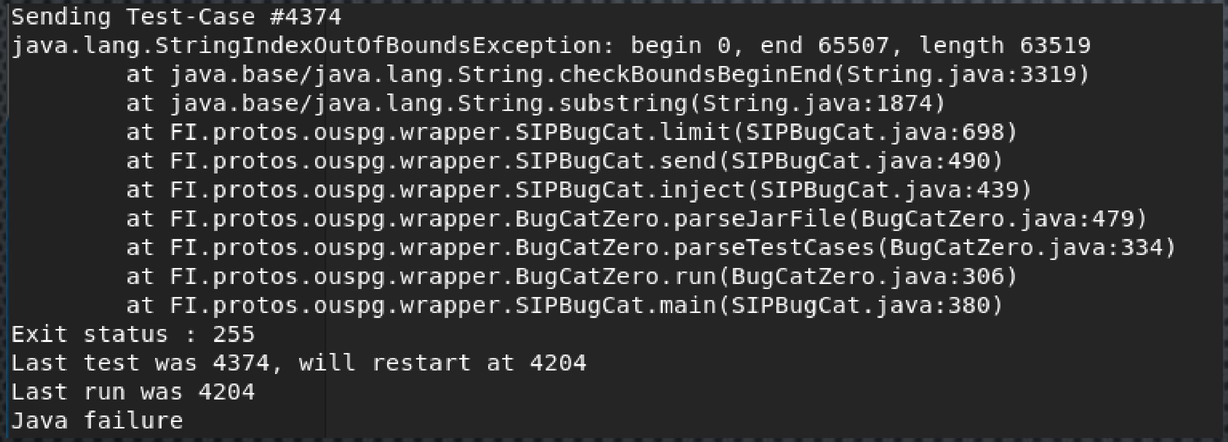
- Old but gold?
- But also buggy ☹

PROTOS DIFFICULTIES

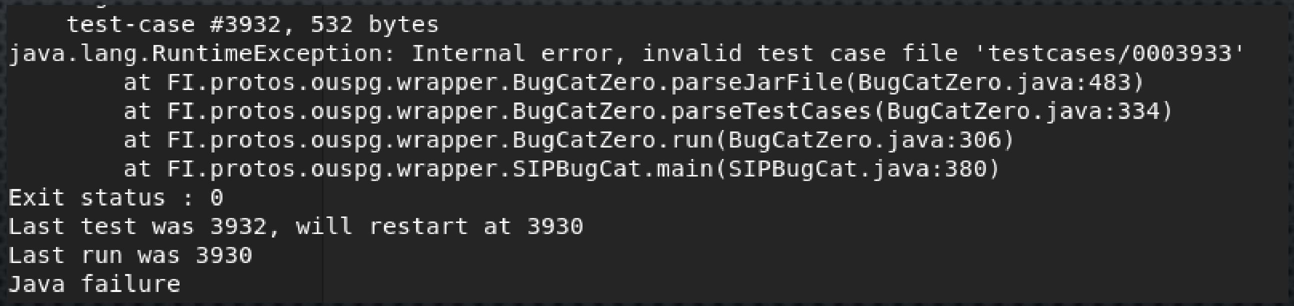
- **Protos** was originally a university project
- **Java** based, it had its glory time in 2003 as a CERT advisory was published impacting multiple PBX vendors
- Evolved as a commercial product

- Original version still usable, nowadays shipped with **KALI** without **documentation**
- Unfortunately it's crashing every few tests (Java errors)
  => Need to **fix** the java code
  OR
  make a script to restart automatically with next test as only some testplans will throw an error.

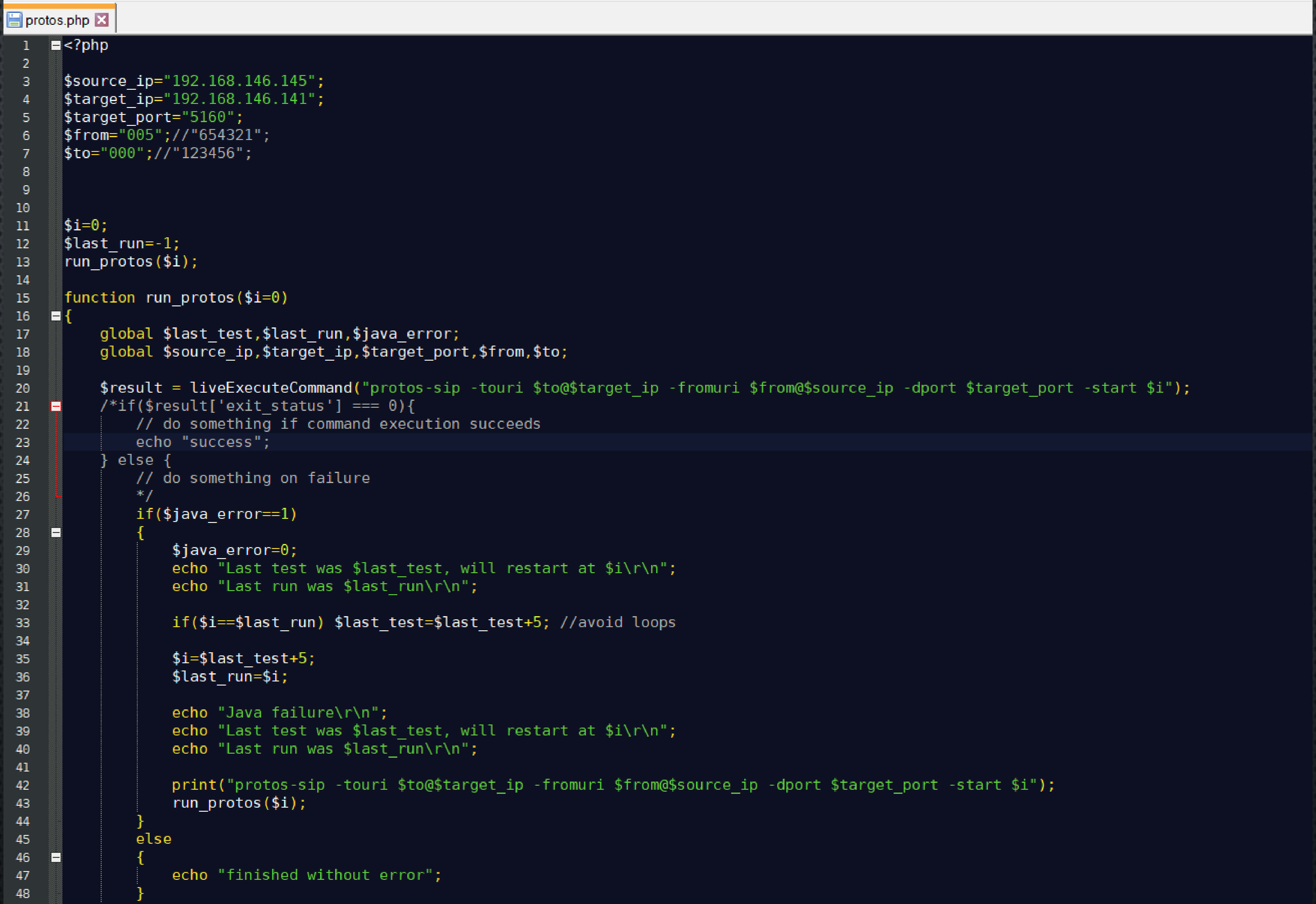
JAVA ERRORS



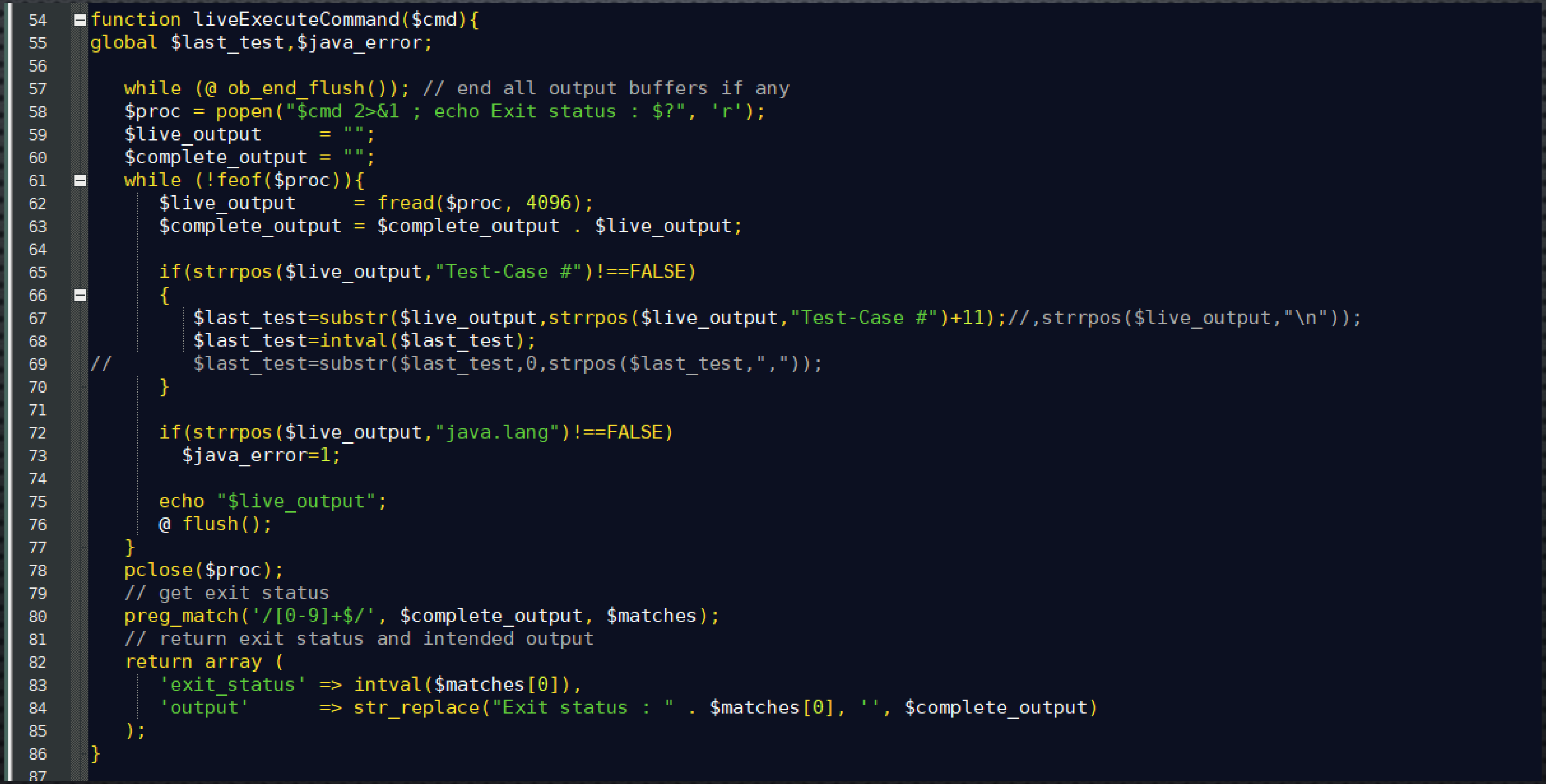
```
Sending Test-Case #4374
java.lang.StringIndexOutOfBoundsException: begin 0, end 65507, length 63519
        at java.base/java.lang.String.checkBoundsBeginEnd(String.java:3319)
        at java.base/java.lang.String.substring(String.java:1874)
        at FI.protos.ouspg.wrapper.SIPBugCat.limit(SIPBugCat.java:698)
        at FI.protos.ouspg.wrapper.SIPBugCat.send(SIPBugCat.java:490)
        at FI.protos.ouspg.wrapper.SIPBugCat.inject(SIPBugCat.java:439)
        at FI.protos.ouspg.wrapper.BugCatZero.parseJarFile(BugCatZero.java:479)
        at FI.protos.ouspg.wrapper.BugCatZero.parseTestCases(BugCatZero.java:334)
        at FI.protos.ouspg.wrapper.BugCatZero.run(BugCatZero.java:306)
        at FI.protos.ouspg.wrapper.SIPBugCat.main(SIPBugCat.java:380)
Exit status : 255
Last test was 4374, will restart at 4204
Last run was 4204
Java failure
```



```
    test-case #3932, 532 bytes
java.lang.RuntimeException: Internal error, invalid test case file 'testcases/0003933'
        at FI.protos.ouspg.wrapper.BugCatZero.parseJarFile(BugCatZero.java:483)
        at FI.protos.ouspg.wrapper.BugCatZero.parseTestCases(BugCatZero.java:334)
        at FI.protos.ouspg.wrapper.BugCatZero.run(BugCatZero.java:306)
        at FI.protos.ouspg.wrapper.SIPBugCat.main(SIPBugCat.java:380)
Exit status : 0
Last test was 3932, will restart at 3930
Last run was 3930
Java failure
```



```php
<?php

$source_ip="192.168.146.145";
$target_ip="192.168.146.141";
$target_port="5160";
$from="005";//"654321";
$to="000";//"123456";



$i=0;
$last_run=-1;
run_protos($i);

function run_protos($i=0)
{
    global $last_test,$last_run,$java_error;
    global $source_ip,$target_ip,$target_port,$from,$to;

    $result = liveExecuteCommand("protos-sip -touri $to@$target_ip -fromuri $from@$source_ip -dport $target_port -start $i");
    /*if($result['exit_status'] === 0){
        // do something if command execution succeeds
        echo "success";
    } else {
        // do something on failure
        */
        if($java_error==1)
        {
            $java_error=0;
            echo "Last test was $last_test, will restart at $i\r\n";
            echo "Last run was $last_run\r\n";

            if($i==$last_run) $last_test=$last_test+5; //avoid loops

            $i=$last_test+5;
            $last_run=$i;

            echo "Java failure\r\n";
            echo "Last test was $last_test, will restart at $i\r\n";
            echo "Last run was $last_run\r\n";

            print("protos-sip -touri $to@$target_ip -fromuri $from@$source_ip -dport $target_port -start $i");
            run_protos($i);
        }
        else
        {
            echo "finished without error";
        }
```



```
function liveExecuteCommand($cmd){
global $last_test,$java_error;

   while (@ ob_end_flush()); // end all output buffers if any
   $proc = popen("$cmd 2>&1 ; echo Exit status : $?", 'r');
   $live_output     = "";
   $complete_output = "";
   while (!feof($proc)){
      $live_output     = fread($proc, 4096);
      $complete_output = $complete_output . $live_output;

      if(strrpos($live_output,"Test-Case #")!==FALSE)
      {
         $last_test=substr($live_output,strrpos($live_output,"Test-Case #")+11);//,strrpos($live_output,"\n"));
         $last_test=intval($last_test);
//       $last_test=substr($last_test,0,strpos($last_test,","));
      }

      if(strrpos($live_output,"java.lang")!==FALSE)
        $java_error=1;

      echo "$live_output";
      @ flush();
   }
   pclose($proc);
   // get exit status
   preg_match('/[0-9]+$/', $complete_output, $matches);
   // return exit status and intended output
   return array (
      'exit_status' => intval($matches[0]),
      'output'      => str_replace("Exit status : " . $matches[0], '', $complete_output)
   );
}
```

VOIPER DIFFICULTIES

- **VoIPER** is a project of 2008 based on **Sulley** fuzzing framework
- Fuzzes SIP/SDP, detects crashes, logs cases
- Also had its glory as it found many crashes in various SIP clients at the time
- presented at **DEF CON** 16

- Still usable today… But also buggy ☹
- Register sequence never succeeded, so what it sent got mostly rejected by PBX
- Sometimes **crashing**, crashing also my Ubuntu lock screen for some reason
- **Workaround**: disable register and have a SIP client on same host registered, e.g Zoiper

WHAT ARE WE LOOKING FOR?

- By sending **unexpected SIP/SDP packets** to the backend, Segmentation **faults** may occur, Buffer **Overflows**, **NULL** pointers, etc..
- All this goes back to improper input validation in the end like XSS/SQLi

=> CRASH?
=> PERFORMANCE ISSUES? (deadlocks, memory leaks, etc)

- Complex protocol means many possibilities out there to try

- conforted by the long list of security advisories https://www.asterisk.org/downloads/security-advisories

EXAMPLE ADVISORIES

- AST-2019-001: Remote crash vulnerability with SDP protocol violation
- AST-2019-002: Remote crash vulnerability with MESSAGE messages
- AST-2018-004: Crash when receiving SUBSCRIBE request
- AST-2018-002: Crash when given an invalid SDP media format description
- …

FIRST FINDING WITH PROTOS

- TEST 1/ Fuzz a PJSIP SIP trunk, just target PBX IP calling the trunk number, no authentication

⇒ approx. 5000 tests, no luck.

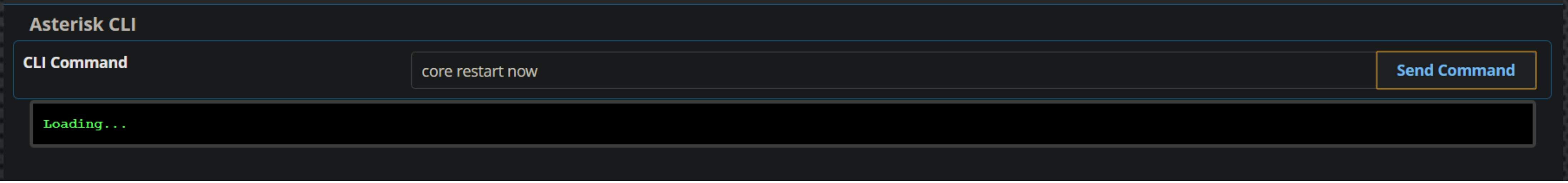
- TEST 2/ Create a PJSIP extension, requires authentication, put 000/000, PB: no auth inbuilt, so registered a SIP client on KALI, Zoiper, then fuzzed Calling from Ext A to Ext B

⇒ approx. 5000 tests, no luck.

- TEST 3/ Create a CHAN_SIP extension, Requires auth, same story. AFTER APPROX 800 tests (1 minute) DEADLOCK !!! No more answers to REGISTER/INVITEs for anyone.

DEADLOCK?

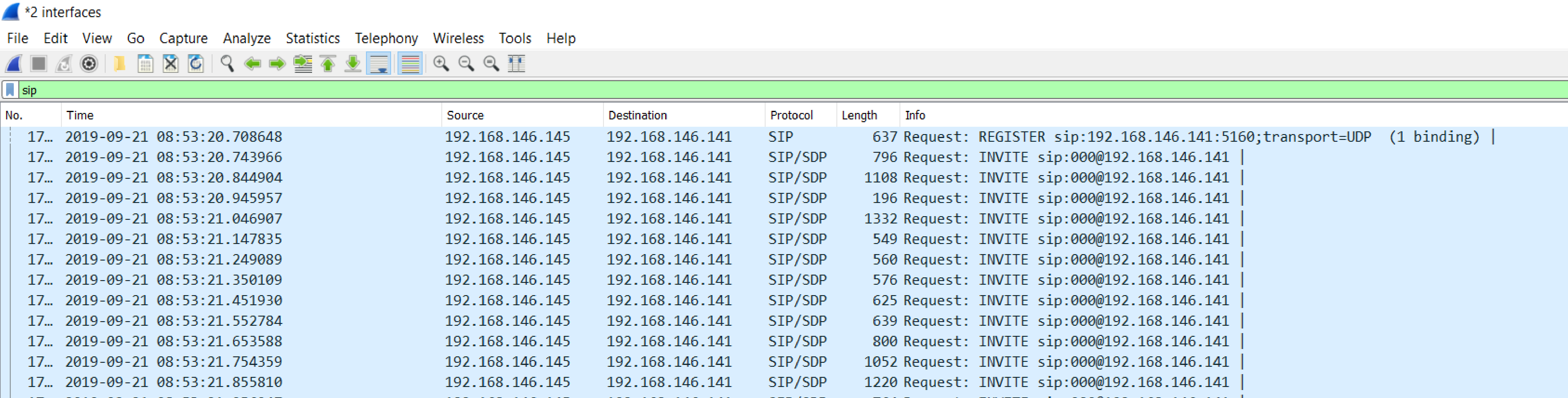
- So, under assault, chan_sip apparently freezed Asterisk 13.22.0
- I verified this wasn't caused by FAIL2BAN as disabled in the first place
- Restarting Asterisk from CLI fixes it !!



Asterisk CLI

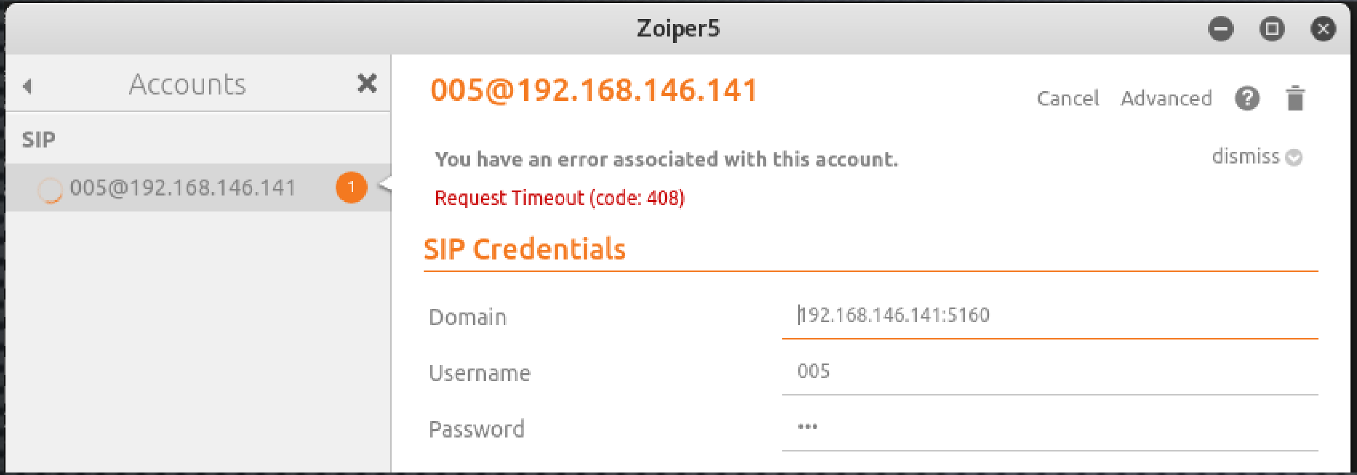
CLI Command: core restart now

```
Loading...
```



Filter: sip

No.	Time	Source	Destination	Protocol	Length	Info
17…	2019-09-21 08:53:20.708648	192.168.146.145	192.168.146.141	SIP	637	Request: REGISTER sip:192.168.146.141:5160;transport=UDP (1 binding) |
17…	2019-09-21 08:53:20.743966	192.168.146.145	192.168.146.141	SIP/SDP	796	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:20.844904	192.168.146.145	192.168.146.141	SIP/SDP	1108	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:20.945957	192.168.146.145	192.168.146.141	SIP/SDP	196	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.046907	192.168.146.145	192.168.146.141	SIP/SDP	1332	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.147835	192.168.146.145	192.168.146.141	SIP/SDP	549	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.249089	192.168.146.145	192.168.146.141	SIP/SDP	560	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.350109	192.168.146.145	192.168.146.141	SIP/SDP	576	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.451930	192.168.146.145	192.168.146.141	SIP/SDP	625	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.552784	192.168.146.145	192.168.146.141	SIP/SDP	639	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.653588	192.168.146.145	192.168.146.141	SIP/SDP	800	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.754359	192.168.146.145	192.168.146.141	SIP/SDP	1052	Request: INVITE sip:000@192.168.146.141 |
17…	2019-09-21 08:53:21.855810	192.168.146.145	192.168.146.141	SIP/SDP	1220	Request: INVITE sip:000@192.168.146.141 |



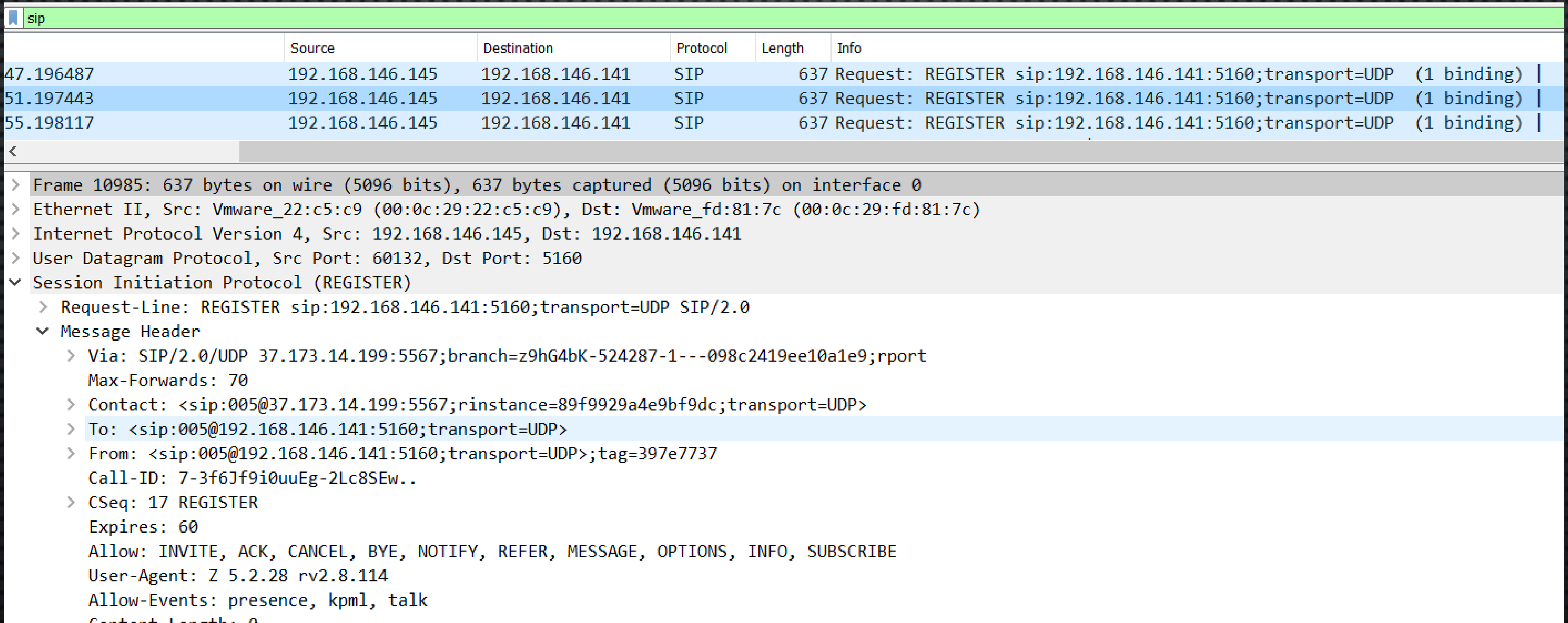
005@192.168.146.141

You have an error associated with this account.

Request Timeout (code: 408)

SIP Credentials

Domain	192.168.146.141:5160
Username	005
Password	•••

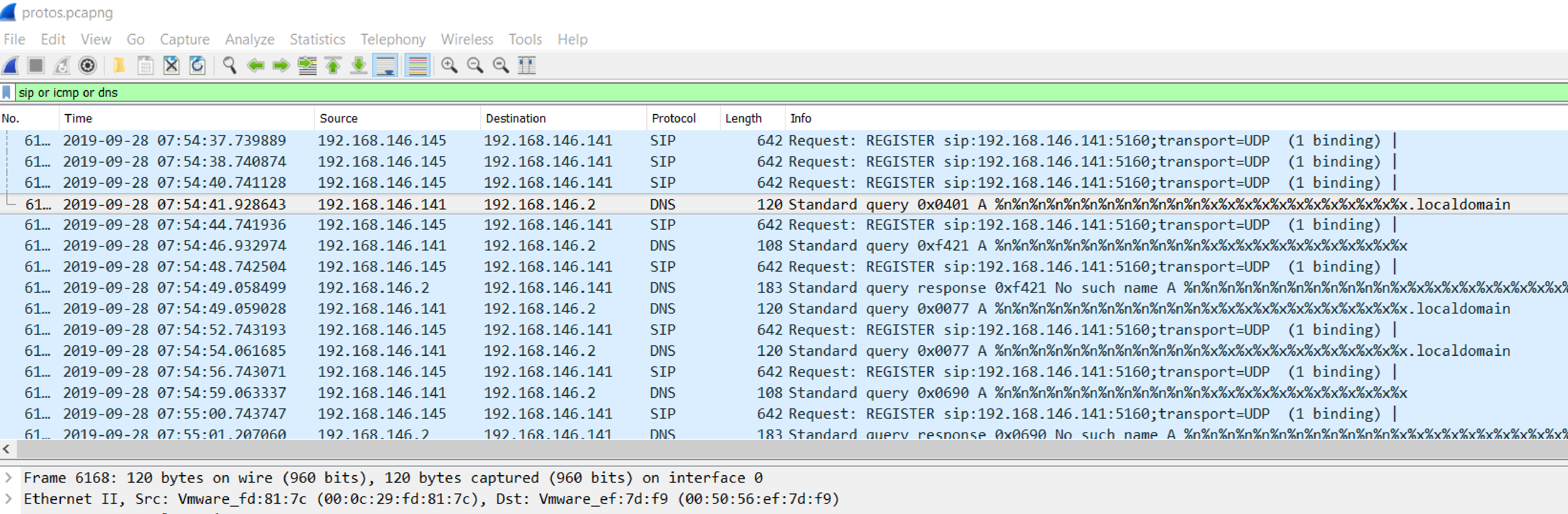


```
             Source            Destination       Protocol  Length Info
47.196487    192.168.146.145   192.168.146.141   SIP          637 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
51.197443    192.168.146.145   192.168.146.141   SIP          637 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
55.198117    192.168.146.145   192.168.146.141   SIP          637 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
```

```
> Frame 10985: 637 bytes on wire (5096 bits), 637 bytes captured (5096 bits) on interface 0
> Ethernet II, Src: Vmware_22:c5:c9 (00:0c:29:22:c5:c9), Dst: Vmware_fd:81:7c (00:0c:29:fd:81:7c)
> Internet Protocol Version 4, Src: 192.168.146.145, Dst: 192.168.146.141
> User Datagram Protocol, Src Port: 60132, Dst Port: 5160
v Session Initiation Protocol (REGISTER)
  > Request-Line: REGISTER sip:192.168.146.141:5160;transport=UDP SIP/2.0
  v Message Header
    > Via: SIP/2.0/UDP 37.173.14.199:5567;branch=z9hG4bK-524287-1---098c2419ee10a1e9;rport
      Max-Forwards: 70
    > Contact: <sip:005@37.173.14.199:5567;rinstance=89f9929a4e9bf9dc;transport=UDP>
    > To: <sip:005@192.168.146.141:5160;transport=UDP>
    > From: <sip:005@192.168.146.141:5160;transport=UDP>;tag=397e7737
      Call-ID: 7-3f6Jf9i0uuEg-2Lc8SEw..
    > CSeq: 17 REGISTER
      Expires: 60
      Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE, OPTIONS, INFO, SUBSCRIBE
      User-Agent: Z 5.2.28 rv2.8.114
      Allow-Events: presence, kpml, talk
      Content-Length: 0
```

DEBUGGING

- Looking further in Wireshark I noticed that Asterisk was still alive trying to resolve bogus **DNS** entries long after (5 Minutes) the fuzz stopped:



protos.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

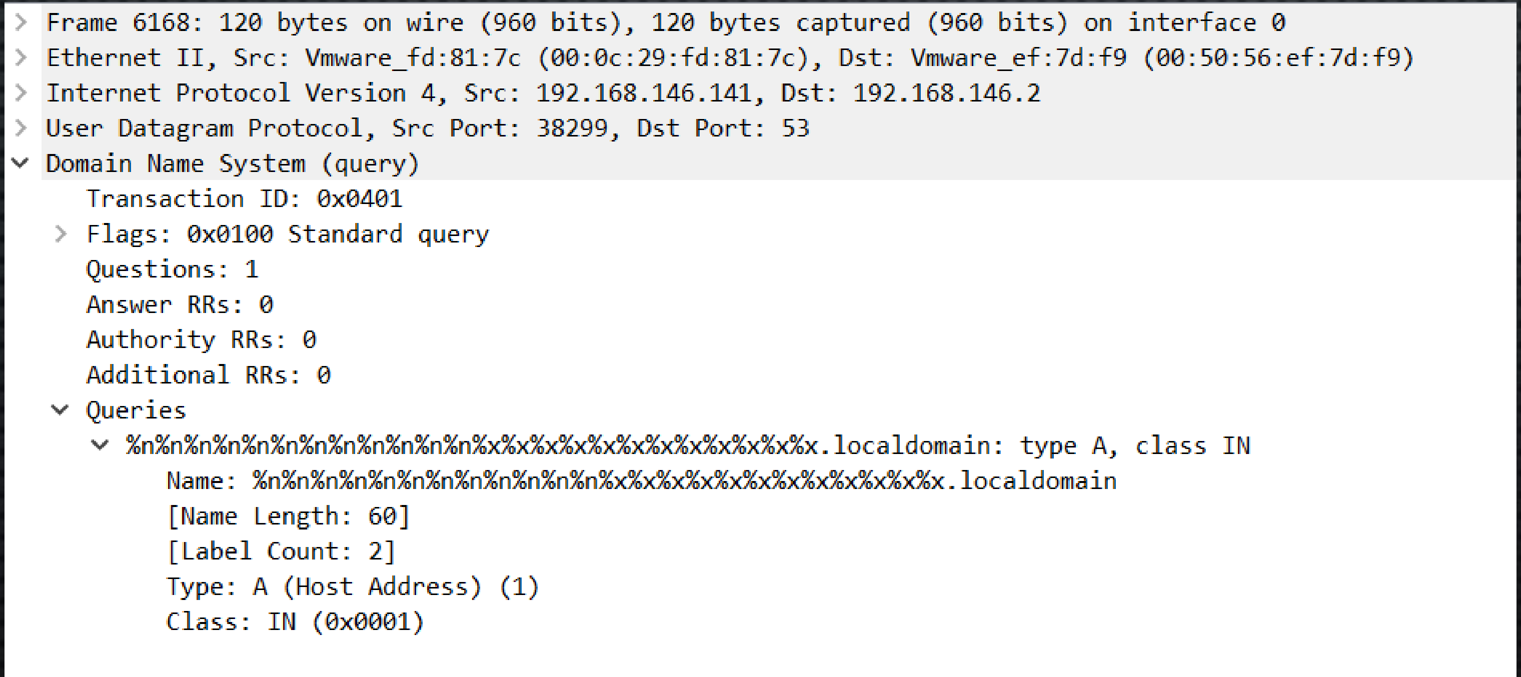
```
sip or icmp or dns
```

```
No.   Time                          Source           Destination      Protocol   Length Info
61…   2019-09-28 07:54:37.739889    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:38.740874    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:40.741128    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:41.928643    192.168.146.141  192.168.146.2    DNS        120 Standard query 0x0401 A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x.localdomain
61…   2019-09-28 07:54:44.741936    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:46.932974    192.168.146.141  192.168.146.2    DNS        108 Standard query 0xf421 A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x
61…   2019-09-28 07:54:48.742504    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:49.058499    192.168.146.2    192.168.146.141  DNS        183 Standard query response 0xf421 No such name A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%
61…   2019-09-28 07:54:49.059028    192.168.146.141  192.168.146.2    DNS        120 Standard query 0x0077 A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x.localdomain
61…   2019-09-28 07:54:52.743193    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:54.061685    192.168.146.141  192.168.146.2    DNS        120 Standard query 0x0077 A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x.localdomain
61…   2019-09-28 07:54:56.743071    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:54:59.063337    192.168.146.141  192.168.146.2    DNS        108 Standard query 0x0690 A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x
61…   2019-09-28 07:55:00.743747    192.168.146.145  192.168.146.141  SIP        642 Request: REGISTER sip:192.168.146.141:5160;transport=UDP  (1 binding) |
61…   2019-09-28 07:55:01.207060    192.168.146.2    192.168.146.141  DNS        183 Standard query response 0x0690 No such name A %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%
```

```
> Frame 6168: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface 0
> Ethernet II, Src: Vmware_fd:81:7c (00:0c:29:fd:81:7c), Dst: Vmware_ef:7d:f9 (00:50:56:ef:7d:f9)
```

DNS REQUESTS NOT ASYNCHRONOUS

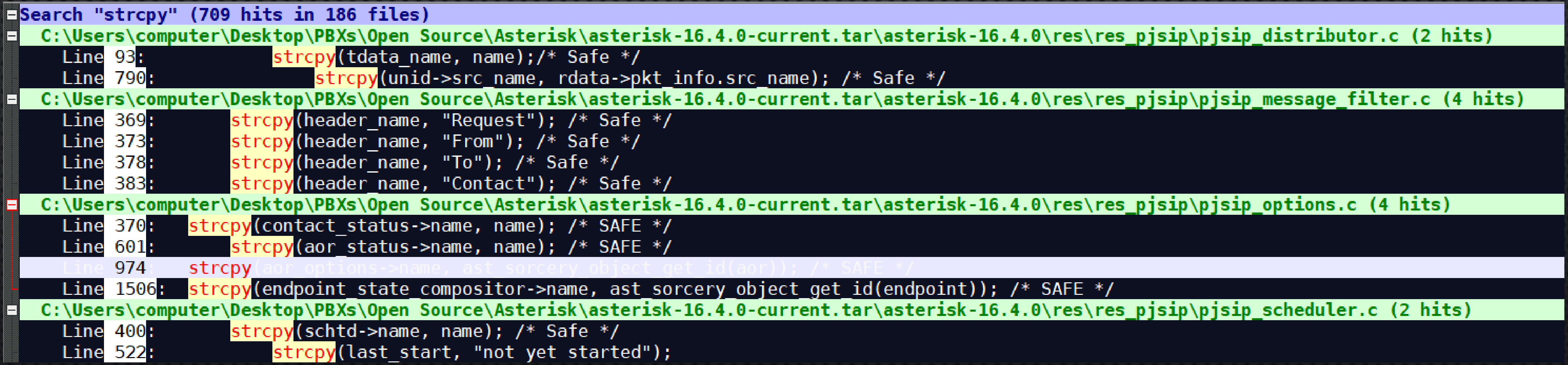
- Digging in Asterisk bug tracker https://issues.asterisk.org/jira/browse
- Turns out it's an old **design issue** in CHAN_SIP, hostnames in SIP packets are looked up in DNS sequentially, so when Protos sends **junk hostname entries** a DNS query occurs.
- Translation: **SIP flood with junk hostnames entries results in CHAN_SIP DoS until all are resolved.**
- Fuzz of 1 minute resulted in my case in 5 minutes downtime of telephony…



```
> Frame 6168: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface 0
> Ethernet II, Src: Vmware_fd:81:7c (00:0c:29:fd:81:7c), Dst: Vmware_ef:7d:f9 (00:50:56:ef:7d:f9)
> Internet Protocol Version 4, Src: 192.168.146.141, Dst: 192.168.146.2
> User Datagram Protocol, Src Port: 38299, Dst Port: 53
v Domain Name System (query)
     Transaction ID: 0x0401
   > Flags: 0x0100 Standard query
     Questions: 1
     Answer RRs: 0
     Authority RRs: 0
     Additional RRs: 0
   v Queries
      v %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x.localdomain: type A, class IN
           Name: %n%n%n%n%n%n%n%n%n%n%n%n%x%x%x%x%x%x%x%x%x%x%x%x.localdomain
           [Name Length: 60]
           [Label Count: 2]
           Type: A (Host Address) (1)
           Class: IN (0x0001)
```

WHAT'S NEXT?

- 1/ CVEs publication
- 2/ fuzz more? Many protocols
- 3/ Audit the C/C++ code?
  - Basic knowledge but can see that plenty SDL BANNED FUNCTIONS are used https://github.com/intel/safestringlib/wiki/SDL-List-of-Banned-Functions
  - e.g searching STRCPY() in latest Asterisk sources found on GitHUB, 709 hits, some marked "SAFE", others not…



```
Search "strcpy" (709 hits in 186 files)
  C:\Users\computer\Desktop\PBXs\Open Source\Asterisk\asterisk-16.4.0-current.tar\asterisk-16.4.0\res\res_pjsip\pjsip_distributor.c (2 hits)
    Line 93:            strcpy(tdata_name, name);/* Safe */
    Line 790:               strcpy(unid->src_name, rdata->pkt_info.src_name); /* Safe */
  C:\Users\computer\Desktop\PBXs\Open Source\Asterisk\asterisk-16.4.0-current.tar\asterisk-16.4.0\res\res_pjsip\pjsip_message_filter.c (4 hits)
    Line 369:       strcpy(header_name, "Request"); /* Safe */
    Line 373:       strcpy(header_name, "From"); /* Safe */
    Line 378:       strcpy(header_name, "To"); /* Safe */
    Line 383:       strcpy(header_name, "Contact"); /* Safe */
  C:\Users\computer\Desktop\PBXs\Open Source\Asterisk\asterisk-16.4.0-current.tar\asterisk-16.4.0\res\res_pjsip\pjsip_options.c (4 hits)
    Line 370:   strcpy(contact_status->name, name); /* SAFE */
    Line 601:       strcpy(aor_status->name, name); /* SAFE */
    Line 974:   strcpy(aor_options->name, ast_sorcery_object_get_id(aor)); /* SAFE */
    Line 1506:  strcpy(endpoint_state_compositor->name, ast_sorcery_object_get_id(endpoint)); /* SAFE */
  C:\Users\computer\Desktop\PBXs\Open Source\Asterisk\asterisk-16.4.0-current.tar\asterisk-16.4.0\res\res_pjsip\pjsip_scheduler.c (2 hits)
    Line 400:       strcpy(schtd->name, name); /* Safe */
    Line 522:           strcpy(last_start, "not yet started");
```

Conclusions

- Seen plenty of examples of vulnerabilities
- caused mainly by improper sanitization of inputs
- Mitigate these by:
  - Following coding good practices (OWASP etc..)
  - E.G standardized parsing, parameterization (SQL)
  - Use of frameworks doing it for you in place of reinventing the wheel
  - Additional controls such as HTTP headers on the webserver level, WAF, etc

- Reporting takes lot of time!

- Latest news, 18/09/2019 Microsoft/GitHub acquires Semmle, a company doing source code auditing software, plans for auto-checks on sources repos?

QUESTIONS?

- Did you like it?
- Anyone wants to contribute next?