

November 2026

PHP2Shell

By Mark Costlow

In the computer world, you know things are bad when a security bug has been given a cutesy name and a logo. There have been a few over the years: **Heartbleed** and **Shellshock** in 2014, **Log4Shell** in 2021, and now **PHP2Shell**. These vulnerabilities only get a name when they are particularly bad. That means they affect software that's used by millions or billions of people worldwide and could be used to wreak havoc if those systems aren't updated. **PHP2Shell** affects all systems using **PHP**, which many people have never even heard of. But PHP is the **programming language** that underpins a huge fraction of the world's web sites, **around 70%** by most estimates.

We were first alerted on October 20th that a **serious PHP bug** was about to be released. These things are given a **severity rating** from 1 to 10, and this one scores 9.2. That means it's Critical, and bad actors can use it to take control of a system without having any special access to it. It's the worst case scenario for a system so deeply entrenched in all of society's computers.

Southwest Cyberport was of course very concerned about this, as almost all of the customer web sites we host use PHP. We spent that night preparing for the release and the patching job ahead. The good news was that the bug disclosure was accompanied by patches for supported versions of PHP within hours. We spent the next days patching systems. Anyone who was already using a relatively recent version of PHP was made safe quickly. The lingering problem was a sizable set of sites still running older versions of PHP. Many of those could be updated easily, but others required code changes to be compatible with the newer versions.

By the time you read this, all sites will have been dealt with. Most will be running on current PHP versions that are patched against this vulnerability, and prepared to update for the next one. Others are running older versions of PHP for which we have back-ported patches. They are safe for now but will need to update to supported versions soon. And some were just too old and required too much work so they have been simply turned off.

Systemic Knowledge Loss

As an aside, this is one way knowledge is forever lost in the Internet age. In ancient times, if someone wrote a book, each copy could be preserved without too much trouble. It has to be protected from sunlight, fire, floods, and pillaging, but nobody needs any special knowledge to do so. If any copy survives, we can make more copies to ensure its continued survival.

In modern times, someone may pour their knowledge and wisdom into a web site, using the best technology of the day, but something like **PHP2Shell** will come along and demand it be updated or lost forever. If the author doesn't have the time, resources, or expertise to update the site, that knowledge is simply lost. There is no "common data format" for the world that can preserve information indefinitely. Since we first began committing our data to computer storage, it has needed constant care and upkeep to preserve not only the data itself, but the systems that surround it to access and manage it.

We are seeing more and more sites created by people who have since passed away. Unless they made special arrangements, their web sites rarely survive the continued funding required to sustain them, and even fewer will survive a situation like this which requires new investment to preserve them. The **Internet Archive** at **archive.org** aims to preserve the entire Internet. They are truly performing a one-of-a-kind service for humanity. Everybody who can should donate a few dollars to keep them going.

AI: Problem or Solution?

PHP2Shell is just the latest high-profile bug discovered with the help of AI. We are in the midst of a revolution where AI-assisted bug hunting is finding critical vulnerabilities at unprecedented speeds and volumes.

Microsoft famously releases patches once a month on "Patch Tuesday". In recent years they have been patching **about 1,000 bugs per year**. There were 1,129 total in 2025, up 11% from 2024. But **since June 2026 they have broken records** almost every month. There were almost 1,000 fixes in September, and over 1,400 in October. That's more in one month than all of 2025 combined!

The good news, if there is any, is that while AI tools are finding these bugs at super-human speed, AI tools are also helping create and release fixes faster than ever. It becomes an arms race, where Microsoft and others are trying to discover, fix, and update these bugs before the criminals can discover and exploit them.

The driving force behind the huge number of Microsoft bug fixes is their project **Codename MDASH**. It's an **Agentic AI system** they developed specifically to find and fix bugs in code. It is specifi-

cally looking for bugs with a security impact, and is adept at finding elusive bugs that might require very specific hard-to-orchestrate conditions to exploit.

The stakes are huge. **UNM Hospital was taken over by cyber-criminals** at the end of October. They apparently had a change of heart when they realized they had disabled life-saving machines and offered the keys to decrypt and restore those systems, but not for the computers making appointments or doing research. We can puzzle over the twisted logic of these villains, but the bottom line is these systems need to be protected from this kind of abuse.

Unfortunately, finding, fixing, and patching a security bug is only a part of the solution. The last step is arguably the most difficult: getting the fix into the running system that needs it. This problem is largely solved phones, laptops, and servers. It's far from figured out for the embedded systems that surround us: TVs, cars, building environmental controls, light bulbs, WiFi routers, refrigerators, and yes, hospital life support machines. Many of these systems do not have established methods and procedures for the people who own them to update them. If they have become out of date, you may be out of luck. There are millions of consumer WiFi routers out in the world that are over 10 years old and haven't had a software update available for a long time. The manufacturer has moved on (if they still exist) so no patches are forthcoming. Eventually the only fix may be to update to a new device, not because old one doesn't work, but because it has become dangerous to have around. That's a real annoyance for a $150 router, but may be a serious industry problem for $10,000 medical machines.

We'll close this section with one more recent infrastructure-level cyber attack. On October 11th someone **hacked into the refrigeration controllers at almost 700 French supermarkets**. They set them to defrost mode, causing them to overheat, and spoiling all of the food. The cost is in the tens of millions of Euros, not to mention the disrupted food supply for many French citizens.

The hacked systems each ran a web server that was reachable over the internet. They were using the manufacturer's questionable password scheme: every one of them used the current date as the password. AI was not needed to find this vulnerability. It was a failure of system design rather than a programming mistake.

This is one of the first infrastructure-level computer security failures we've seen, but it will not be the last. In fact, on November 2nd, 3 weeks after the French attacks, grocery workers in Belgium, Sweden, Finland, Ireland, and Estonia all **found their freezers had intentionally spoiled their food**.

Many systems use the "security by obscurity" method - the mistaken belief that nobody will guess the esoteric method used to access the system so it's safe from outsiders who haven't been trained on it. This is a case where AI didn't cause the problem, but it could certainly have prevented it. If AI is used in the deployment process, to evaluate the system plan, it can insist on best practices and flag things like bad password management before it gets put out into the world.

The **Shodan** web site is a **search engine for devices and services exposed** to the internet. For example, it tells us there are almost 2 million MySQL servers open to probes from the internet. MySQL is a database server used by web sites, and should always be hidden from direct Internet login, but 2 million of them are exposed, waiting for AI to probe their flaws.

Steve Gibson, long-time software developer and host of the **Security Now podcast**, believes we are in the middle of an upheaval that will leave us with better software than ever. I tend to agree that in a few more months the majority of "current" code on the internet will be of higher quality, with fewer bugs than ever. But that doesn't address the pockets of old code that may never get updated until it gets abused in some way and wakes someone up to its existence. Lets hope we survive long enough to see that bug-free promised land.

What Can We Do?

There's not a lot we as individuals can do to stop the AI juggernaut. It's simply too useful and powerful to talk people out of using it. But one vulnerability we should put the brakes on is embedding Internet access in every appliance. There is no good reason for a home fridge, oven, or dishwasher to need an app. When you buy an appliance, insist on a model that does not require an app to function. As we'll see in a future issue, many of those apps are far more sinister than they appear. We shouldn't aid them at the expense of our privacy and security.

This Month in Ideas & Coffee

Check out the free talks and classes this month!

- **November 11 6pm - 8pm** – Building a WordPress Website with Claude
- **November 17 6pm - 7:30pm –** *WordPress Work Along* – Answers to your pressing WordPress questions. Bring your laptop!

Watch the **Ideas & Coffee** event calendar at **https://swcp.com/calendar** for future info.

Southwest Cyberport – swcp.com

New Mexico's Expert Internet Service Provider since 1994

505-232-7992 | Support: help@swcp.com

5021 Indian School NE, Ste. 600, Albuquerque, NM 87110

Click on bold blue type in browser for links.