Incident Report – WordPress Server Compromise

Date: 2025-02-01

Handlers: John H Lee, Prakash Kanchan, Suraj Khadka

Executive Summary

On February 9th, 2021, at 6:05 AM ET, Department XYZ notified the Georgia Tech Security Operations Center (SOC) about their research blog showing unfamiliar contents. A three-person Incident Response (IR) team was assigned to investigate, who disconnected the WordPress host from the network and uncovered evidence of a breach – consisting of a Remote Code Execution (RCE) attack, several malicious files, and a compromised SQL database. Further analysis of the compromised device and network traffic indicated the damage was limited to website defacement. The IR team worked with SOC, Department XYZ and IT to rebuild impacted systems from a clean backup and applied security upgrades for future use. Normal blog operations resumed on February 11th, 2021, at 8:00 AM ET. The operational impacts of this breach were minor, only disrupting Department XYZ's research blog for 5 days. Total financial impact to the organization is assessed at $6,200.

Background

On February 2nd, 2021, Department XYZ created a blog to showcase research in their field. They hosted this website on Georgia Tech infrastructure, selecting WordPress, a popular open-source Content Management System (CMS), to serve their content. The latest version of WordPress available at the time (v5.6.1) and its associated plugins contained several vulnerabilities[1]. Some of these vulnerabilities were successfully probed and exploited by the attacker.

WordPress is backed by a SQL Database (DB) which stores blog data such as posts and comments. However, the DB also stores user authentication information, when password-based authentication is implemented. Owing to urgency around the research being shared, Department XYZ deferred IT's recommendations on setting up Single Sign On[2] (SSO) and a demilitarized zone

[1] "WordPress 5.6.1 Vulnerabilities." 2024. WPScan. 2024. https://wpscan.com/wordpress/561/

[2] Teravainen, Taina. "What Is Single Sign-on (SSO) and How Does It Work?" Search Security, April 10, 2024. https://www.techtarget.com/searchsecurity/definition/single-sign-on

(DMZ) architecture[3]. Instead, they opted to use password authentication - hosting the Webserver and DB server on the same machine. This allowed the attacker to exploit credentials on the SQL DB after gaining Remote Code Execution (RCE) access to the webserver.

Furthermore, Department XYZ also deferred the recommended integration with Georgia Tech's Endpoint Protection Platform (EPP), Network Monitoring, and Security Information and Event Management (SIEM) system - allowing suspicious activity to go undetected.

Over the course of four days (February 3rd – February 6th, 2021), malicious actors compromised this WordPress blog. The attackers dropped several malware files onto the server, performed RCE, and compromised admin credentials in the DB. Armed with these credentials, the attackers were able to modify blog posts – overwriting the author's posts with their own.

On February 9th, 2021, at 6.05AM ET, Department XYZ noticed their blog's defacement and called the Georgia Tech SOC HelpDesk number (1-888-GIT-HELP) to report this incident. The SOC quickly assigned a team of 3 researchers to investigate and remediate this incident.

The IR team immediately disconnected the WordPress host from the network, taking the blog offline. Next, the IR team carefully preserved the state of the server by snapshotting its memory and disk. The team then started forensically investigating access logs, the file system and the WordPress database for indicators of compromise (IoCs).

Timeline

Note: *All timestamps are in 24-hour format, Eastern Standard Time*

- **Feb 3, 2021** – from IP **`47.75.76.54`**, registered at Alibaba Cloud[4]
  - 09:32:40 – An attacker attempts to access `xmlrpc.php` on Department XYZ's WordPress site, which is blocked with a 405 error.
  - 09:32:41 – The attacker opens the WordPress login page.
  - 09:32:42 – The attacker attempts the user enumeration exploit[5]
  - 09:32:43 – User enumeration exploit succeeds, WordPress returns the admin author page, exposing admin username.

[3] Wikipedia. "DMZ (Computing)." Wikipedia, January 11, 2025. https://en.wikipedia.org/wiki/DMZ_(computing)
[4] "IP Address Lookup for 47.75.76.54 in Hong Kong, Hong Kong." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/47.75.76.54
[5] Park, Bhagwad. 2016. "Here's How Hackers Can Find Your WordPress Username." WP-Tweaks. September 9, 2016. https://www.wp-tweaks.com/hackers-can-find-your-wordpress-username/

- 09:32:46 – The attacker kicks-off over 800 brute-force calls[6] to exploit an `xmlrpc.php` vulnerability and reveal the admin password.
- 09:42:03 – The brute force attacks end. It appears the brute-force was unsuccessful, because there is no attempted login following this.

- **Feb 4, 2021** – from IP `72.167.247.190`, registered at GoDaddy.com[7]
  - 14:23:08 – Attacker attempts to gain Remote Code Execution (RCE) via the `timthumb.php` exploit[8] with a maliciously constructed URL `img.youtube.com.dollhousedelight(dot)com/.mods/bbb.php`, looking for the file in several theme and plugin folders.
  - 14:23:13 – After 4 failed attempts, the attacker succeeds in exploiting `timthumb.php` found inside `wp-content/plugins/wordpress-gallery-plugin`, uploading an apparent web shell (`bbb.php`) to the directory
  - 14:23:18 – Attacker confirms exploit by issuing a command via the `bbb.php` web shell and succeeds.

- **Feb 5, 2021** – from IP `72.167.34.8`, registered at GoDaddy.com[9]
  - 09:12:01 – Attacker returns with a new IP and issues 5 commands to `bbb.php`. The logs do not indicate the extent of activity performed here, but one or several of these calls created (and modified) the malicious file with the "bloodninja" exploit[10] - "`wp-update.php`" which has a last modified timestamp of "Feb 5, 2021, 09:14:00".

- **Feb 5, 2021** – from IP `72.167.31.2`, registered at GoDaddy.com[11]
  - 09:14:56 – Attacker returns with a new IP and makes a call to `bbb.php`. The logs do not indicate the extent of activity performed here.
  - 09:15:45 – Attacker makes a call to `bbb.php`. This call created/modified the malicious `cnrig` crypto miner persistence file `./wp-includes/e4uicklw47.php`, which has the same exact last modified timestamp.

---

[6] "Exploiting the Xmlrpc.php on All WordPress Versions." 2019. Lucian Nitescu. July 2019. https://nitesculucian.github.io/2019/07/02/exploiting-the-xmlrpc-php-on-all-wordpress-versions/#brute-force-attacks

[7] "IP Address Lookup for 72.167.247.190 in Tempe, United States." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/72.167.247.190

[8] https://www.facebook.com/priteshvora1982. 2023. "How to Remove TimThumb Hack from WordPress Website." MalCare. October 9, 2023. https://www.malcare.com/blog/wordpress-timthumb/

[9] "IP Address Lookup for 72.167.247.190 in Tempe, United States." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/72.167.247.190

[10] Patil, Vishal. 2019. "Cleaning Bloodninja PHP Attack from Wordpress Website." Medium. InfoSec Write-ups. May 11, 2019. https://infosecwriteups.com/cleaning-bloodninja-php-attack-from-website-7703513bd133

[11] "IP Address Lookup for 72.167.247.190 in Tempe, United States." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/72.167.247.190

- 09:16:30 – Attacker makes a call to `bbb.php`. This call created/modified the malicious web-shell[12] file `./wp-content/themes/twentytwenty/403.php` which has the same exact last modified timestamp.
- 09:20:32 – Attacker issues 2 commands via `bbb.php`. The logs do not indicate the extent of activity performed here.

- **Feb 6, 2021** – from IP `157.75.167.23` registered to JR East Information Systems Company[13]
  - 23:39:49 – Attacker returns with a new IP to and makes 2 calls to `bbb.php`. The logs do not indicate the extent of activity performed here.
  - 23:40:34 – Attacker makes a call via `bbb.php`. This is where the attacker added a second admin user "`admin2`" into the DB, likely obtaining its credentials from the `wp-config.php` file. This activity correlates with the `user_registered` timestamp on `wp_users`.
  - 23:40:56 – Attacker runs a final command through `bbb.php`. This is likely where the attacker deleted `bbb.php` from the cache, because it is no longer on the system.
  - 23:41:23 – Attacker attempts and successfully logs in to the WordPress UI as `admin2` and now has full access to the admin site.
  - Between 23:41:24 and 23:46:04 - we see several interactions logged between the attacker and WordPress, ending in the "Buy Essay Papers" post being last updated at 23:43:56, as per the DB timestamp.
- **Feb 9, 2021**
  - 06:05:00 – Department XYZ contacts SOC.
  - 06:15:00 – SOC creates **INC001234** and assigns it to IR team.
  - 06:25:00 – IR team assembles team of 3 researchers - John H Lee, Prakash Kanchan, Suraj Khadka – to analyze the incident.
  - 10:30:00 – IR team takes the WordPress host offline.
  - 11:00:00 – IR team acquires access logs, DB dump, and web server directory archives for forensic analysis.
  - 17:00:00 - IR team concludes initial investigative work.
- **Feb 10, 2021**
  - 06:40:00 – IR team compiles their findings on the breach, along with recommendations to setup and secure a new WordPress instance before making it operational for Department XYZ
  - 10:00:00 – IR team engages IT personnel to setup new WordPress instance with necessary remediations.
  - 12:00:00 – IT begins implementing SSO authentication, host and network security measures.
  - 14:00;00 - IT makes WordPress application remediations and updates.
  - 16:00:00 - IR team receives confirmation from IT that remediations are complete.

---

[12] NavyTitanium. 2020. "Misc-Malwares/Emotet/Webshells/Menu.php at Master · NavyTitanium/Misc-Malwares." GitHub. 2020. https://github.com/NavyTitanium/Misc-Malwares/blob/master/Emotet/webshells/menu.php

[13] "IP Address Lookup for 157.75.167.23 in Tokyo, Japan." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/157.75.167.23

- **Feb 11, 2021**
    - 07:00:00 – IR supervises scans of the servers and validates remediations.
    - 08:00:00 – Under supervision of IR team, IT restores network connectivity, and blog is operational.
    - 09:00:00 - INC001234 is closed.

Findings

Summary:

- Using vulnerable WordPress plugins allowed RCE, and injection of malware files
- Hosting Webserver and DB server on the same host, combined with the use of password-based authentication, allowed for credential compromise and eventual content overwrites
- Not following Georgia Tech best practices on logging and monitoring, allowed the breach to go undetected

Details:

Note: *All times in Eastern Standard Time*

On February 2nd, 2021, around 8:32 AM, Department XYZ submitted a request to IT to create a blog to share some urgent research with the public. Email exchanges between IT and XYZ around 9:00 AM show that IT recommended setting up SSO, DMZ and onboarding to Georgia Tech's EPP and SIEM, but XYZ wanted the website to go live "ASAP".

On February 2nd, 2021, at 12:32 PM, IT created the blog as requested by Department XYZ and deferred SSO/DMZ setup and EPP/SIEM onboarding. XYZ immediately made their first research blog post, and the website was live on the internet.

On February 3rd, at 9:32 AM, the first set of attacks began. Using an IP registered at Alibaba Cloud[14], the attacker first tried a WordPress user enumeration exploit[15] to glean the admin username. Next, the attacker launched over 800 calls to the vulnerable `xmlrpc.php` endpoint in a brute-force attempt to break the password for the admin account. This file has a known exploit[16] where a hacker can use the `wp.getUsersBlogs` method, to send a username/password pair, and the endpoint reveals whether the password was correct. Analysis indicates that these brute-force attempts failed, and the attacker gave up after 10 minutes.

---

[14] "IP Address Lookup for 47.75.76.54 in Hong Kong, Hong Kong." 2019. WhatIsMyIPAddress. 2019. https://whatismyipaddress.com/ip/47.75.76.54

[15] Park, Bhagwad. 2016. "Here's How Hackers Can Find Your WordPress Username." WP-Tweaks. September 9, 2016. https://www.wp-tweaks.com/hackers-can-find-your-wordpress-username/

[16] 0xlucifer. (2024, February 18). *Wordpress Brute Force attack*. Medium. https://medium.com/@0xlucifer/wordpress-brute-force-attack-7b5b5054cee9

On February 4th, around 2:23 PM – the attacker returned with another IP registered at GoDaddy.com. This time the attacker probed for vulnerable plugins containing the file "`timthumb.php`"[17], finally locating it under the "`wordpress-gallery-plugin`" path. `timthumb.php` is intended to download image thumbnails from various websites but uses an inadequately constructed filter to allow list websites[18] - `img.youtube.com` being one of them. Using the URL `img.youtube.com.dollhousedelight(dot)com/.mods/bbb.php`, the attacker was able to download the presumed web shell: `bbb.php` into the cache and achieve unprivileged RCE on the webserver by invoking it.

On February 5th, around 9:14 AM, the attacker returned with a second IP registered at GoDaddy.com and made 5 calls to `bbb.php`. The logs do not indicate the full extent of activity performed here, but a series of calls created the malicious file "`wp-update.php`" containing the "bloodninja" exploit[19] , which can redirect traffic to a malicious site via an injected cookie. The investigation indicated that this file never got invoked before the IR team shut down the XYZ server.

On February 5th, around 9:14 AM, the attacker returned with a third IP registered at GoDaddy.com and made a call to `bbb.php`. The logs do not indicate the extent of activity performed here.

On February 5th, around 9:15 AM, the attacker used `bbb.php` to create "`e4uicklw47.php`", an apparent persistence file for the `cnrig` crypto miner[20] which runs commands to "donate" 40% CPU for 1 minute every 100 minutes to a mining pool hosted at "`pool.aeon.hashvault.pro`". Analysis indicates that this file never got invoked before the IR team shut down the XYZ servers and the `cnrig` binary itself was not located on the WordPress host.

On February 5th, around 9:16 AM, the attacker utilized the `bbb.php` file to drop a web shell[21] file `403.php`. This file never got invoked before the IR team shut down the XYZ servers. The attacker then made two more calls `bbb.php`. The logs do not indicate the extent of activity performed here.

On February 6th, around 11:39 PM, the attacker returned with an IP registered at JR East Information Systems Company and made 3 more calls to `bbb.php`. One of these calls added a second admin user "`admin2`" into the DB. The attacker was likely able to log into the DB using plain-text credentials from the `wp-config.php` file. A final command was then issued to `bbb.php` which likely removed the `bbb.php` file itself from the cache, as it did not exist in the

---

[17] https://www.facebook.com/priteshvora1982. 2023. "How to Remove TimThumb Hack from WordPress Website." MalCare. October 9, 2023. https://www.malcare.com/blog/wordpress-timthumb/

[18] NVD. "You Are Viewing This Page in an Unauthorized Frame Window." NVD, 2011. https://nvd.nist.gov/vuln/detail/CVE-2011-4106.

[19] Patil, Vishal. 2019. "Cleaning Bloodninja PHP Attack from Wordpress Website." Medium. InfoSec Write-ups. May 11, 2019. https://infosecwriteups.com/cleaning-bloodninja-php-attack-from-website-7703513bd133

[20] "CNRig." 2022. GitHub. November 14, 2022. https://github.com/cnrig/cnrig

[21] NavyTitanium. 2020. "Misc-Malwares/Emotet/Webshells/Menu.php at Master · NavyTitanium/Misc-Malwares." GitHub. 2020. https://github.com/NavyTitanium/Misc-Malwares/blob/master/Emotet/webshells/menu.php

forensic snapshot of webserver files. Finally, one minute later, the attacker logged in as an admin to the XYZ blog via the UI and made several edits. One of these edits modified the original blog post from Department XYZ – changing it to show the "Buy Essay Papers" message instead.

No further activity was observed by the attacker.

Actions Taken

Immediately upon notification of a compromise on the morning of 9th February 2021, the SOC initiated an incident, **INC001234**, and assembled a three-member IR team to assess the possible breach. After receiving an initial briefing on the situation from department XYZ and IT teams, the researchers accessed the WordPress host and disconnected the device from the network. Using forensic tools, the IR team took a snapshot of the server disk and memory to preserve all IoCs.

The IR team subsequently analyzed webserver access logs and the exploited WordPress application itself, including its backing SQL database - as primary sources of evidence. Utilizing log aggregation tools such as Splunk and leveraging open-source research on sites such as GitHub, ExploitDB, and VirusTotal, the IR team pieced together a timeline of events during the attack.

Investigating this sequence of events allowed the team to confirm that:

- The breach was restricted to only a blog post being modified
- The threat actor had not launched any of the other malware files besides "`bbb.php`"
- The threat actor had not established any persistent access on Georgia Tech networks or succeeded in lateral propagation outside the WordPress host
- No other containment/eradication was necessary beyond taking the host offline and purging it.

Following their investigation, the IR team engaged with - the SOC, department XYZ's website owners, and the IT department to initiate a remediation plan. All relevant stakeholders agreed on the plan and IT began the recovery process at noon on February 10th.

Remediation included the following steps:

1. Set up new WordPress instance, with web server on a DMZ *(initially without external networking)*, and Database on a separate server hosted behind a secure firewall.
2. Restore the compromised data from a clean backup (taken prior to the intrusion and dated February 2nd, 2021)
3. Change WordPress authentication from username and password to full SSO integration with the Georgia Tech IDP; and enable multi-factor authentication.
4. Integrate the webserver and DB hosts with Georgia Tech EPP, which uses puppet to install a host-based antivirus solution, Sophos Protection for Linux.

5. Add network-based protection measures by placing the web server behind a separate, reverse-proxy device that includes a web-application firewall which inspects and blocks suspicious traffic.
6. Configure log forwarding to a centralized Splunk SIEM managed by the SOC.
7. Fix WordPress vulnerabilities:
    a. Disable "`xmlrpc.php`" access via `.htcaccess`[22]
    b. Eliminate all plugins using "`timthumb.php`", and instead use plugins with secure alternatives like mThumb[23]
    c. Patch `admin-ajax.php` to block SQL Injection[24]
    d. Restrict WordPress Admin access via `.htcaccess`, to only Georgia Tech IPs[25]
    e. Enable WordPress vulnerability scanner "`wpscan`" via EPP and kick off initial scan to confirm a vulnerability-free setup.

The remediation plan was verified complete by the IR team on February 11th. Consequently, network access to the WordPress servers was restored and the blog was live and operational at 8:00 AM ET, February 11th – following which the Incident **INC001234** was marked "Resolved".

Financial Impact

Item	Cost
Lost Business and Productivity Costs [1]	$200
Labor[2]	$6,000
Total	**$6,200**

1. Website is associated with non-profit research, but we assess an advertising revenue loss of $200 for the ~5-day period between blog defacement and remediation
2. Total labor cost of incident response was calculated as 20 investigative hours x $100 per investigative hour x 3 personnel = $6000

[22] MacLeod, Rianna. 2023. "What Is XML-RPC? Security Risks & How to Disable." Sucuri Blog. May 4, 2023. https://blog.sucuri.net/2023/05/what-is-xml-rpc-security-risks-how-to-disable.html
[23] mindsharelabs. 2016. "GitHub - Mindsharelabs/Mthumb: MThumb - a Secure TimThumb Alternative for Easily Resizing Images." GitHub. February 22, 2016. https://github.com/mindsharelabs/mthumb
[24] Gohil, Pathik. 2022. "Understanding the WordPress SQL Injection Vulnerability (CVE-2022-21661) - Vsociety." Vicarius.io. 2022. https://www.vicarius.io/vsociety/posts/understanding-the-wordpress-sql-injection-vulnerability-cve-2022-21661
[25] "[DIY] Learn How to Restrict Access to Wp-Admin in 5 Minutes." 2020. Astra Security. March 31, 2020. https://www.getastra.com/blog/wordpress-security-course/restrict-access-to-wp-admin/

Lessons Learned

Successes

- Timely engagement of SOC on identification of defacement by department XYZ prevented opportunities for the attacker to cause damage beyond blog modifications, such as establishing persistent access or lateral movement into Georgia Tech network.
- Tight collaboration between the IR team, SOC, and IT ensured a structured remediation process - significantly reducing the risk of a future WordPress attack.
- The transition to a hardened WordPress application with SSO integration, database isolation and endpoint protection was completed in 48 hours by SOC team and IT - limiting the breach's financial impact to $6,200, with minimal data loss, and minimal reputational damage.

Opportunities for Improvement

Issue: Lack of consistency in implementing strong authentication
Recommendation: Update the Information Security Policy (ISP), to *require* implementation of Single-on (SSO) via Georgia Tech IDP, for all applications – even when applications are classified "Public". This ensures consistency in security and prevents breaches originating from applications relying on password authentication.
Action Item Owner: Information Security Manager (ISM)

Issue: Lack of consistency in Network and Host-Based Monitoring
Recommendation: Inventory all Georgia Tech servers, and onboard them to the EPP. This will enable real-time monitoring for suspicious login attempts, web traffic anomalies, and unauthorized modifications of files
Action Item Owner: SOC Team

Issue: Lack of application allowlist.
Recommendation: Update the ISP with a concise list of applications that can be hosted on Georgia Tech infrastructure. WordPress should be discouraged because it is rife with vulnerabilities. Alternatively, a Georgia Tech fork of WordPress could be created and hosted internally, which has all the security upgrades baked in, and with tenants for each department's use cases.
Action Item Owner: Information Security Manager (ISM)

Issue: Use of Vulnerable WordPress Plugins
Recommendation: Create an allowlist of permitted WordPress plugins. Regularly audit and patch plugins to ensure only secure plugins are installed.
Action Item Owner: SOC Team

Issue: Lack of mandatory application review by Cybersecurity teams before launch
Recommendation: Update the ISP to require SOC review/approval of all applications, irrespective of data classification – before go-live. This can catch business vetoes of IT that may introduce cyber risk for Georgia Tech
Action Item Owner: Information Security Manager (ISM)

ASSUMPTIONS MADE BY GROUP 25 FOR THIS ASSIGMENT

In order to present a more complete picture of the incident and remediation, we've made the following assumptions

- **Incident site**
  - One of the reflected IPs in logs pointed to a Georgia Tech host, so we're assuming this entire incident occurred at Georgia Tech
- **Organization**
  - Department who asked for the blog is called "XYZ"
  - Department XYZ had veto authority over this IT
  - Incident Response team belongs to SOC
  - SOC has a helpdesk for reporting incidents 1-888-GIT-HELP
- **Infrastructure/Cybersecurity**
  - **No other files were modified besides what we were provided with in the scope of this assignment** (aka files outside the Wordpress folder were not modified, because we have no access to those files in the assignment)
  - Some of the malware base64 strings were obfuscated partially. We assume this was to avoid students running the site locally and triggering an exploit. We've searched online for the non-obfuscated strings, and used those as markers of what the original malware may have been.
  - Initial WordPress setup had the DB and Webserver on the same host machine
  - Georgia Tech already has centralized IDP, SIEM and EPP systems - and tools such as Sophos, puppet, Splunk, wpscan
  - Georgia Tech already has a DMZ network design
  - IT recommended integration into these but were vetoed because of urgency around research papers
  - ISP document was missing mandating SSO, which allowed this blog to go online with password auth
  - Infrastructure inventory was missing this WordPress server, allowing it to go undetected and unmonitored
  - There wasn't an allowlist of applications or plugins, allowing a vulnerable WordPress setup to go ahead
- **Others**
  - Incident was given a ticket number INC001234
  - Incident response team charges $100/hour per resource, and it took 20 hours of effort each, from 3 resources, to address this incident
  - Advertising revenue is the only loss for this blog, estimated at $200