

The Bondnet Army | Akamai

akamai.com/blog/security/the-bondnet-army

The Bondnet Army

Written by

[Daniel Goldberg](#)

June 08, 2020

Written by

[Daniel Goldberg](#)

Daniel is a security research expert at Guardicore Labs, where he is responsible for tracking the latest security intelligence, including detailed analysis of hackers' methodologies, for use in creating advanced countermeasures for Guardicore products and services. Daniel has more than 10 years of cybersecurity research experience.

Share

Guardicore Labs has recently picked up Bondnet, a [botnet](#) of thousands of compromised servers of varying power. Managed and controlled remotely, the Bondnet is currently used to mine different cryptocurrencies and is ready to be weaponized immediately for other purposes such as mounting [DDoS attacks](#) as shown by the [Mirai Botnet](#). Among the botnet's victims are high profile global companies, universities, city councils and other public institutions.

The attacker behind Bondnet breaches the victims through a variety of public exploits and installs a Windows Management Interface (WMI) trojan that communicates with a Command and Control (C&C) server. Operating under the name Bond007.01, the attacker can then take full control of the servers to exfiltrate data, hold it for ransom, use the server to stage further attacks and more. Active since December 2016, Bondnet primarily mines [Monero](#). Bond007.01 is financially motivated, earning around a thousand dollars a day.

Bondnet by Numbers

- Bondnet has penetrated more than 15,000 machines to date
- Everyday 2000 machines – equals 12,000 cores – report to the Bondnet C&C

- Victim CPU core count varies from 1 to 64
- ~ 500 new machines are added daily to the attacker's network and around the same number of machines is delisted
- Bondnet victims are distributed across 141 countries in 6 continents

Botnet Recruitment

The initial attack coming from Hong Kong exploited a known weak configuration in phpMyAdmin that allowed the attacker to deploy unknown DLLs and an encoded Visual Basic script. None of these files were detected by the multiple Antivirus deployed on the botnet's victims nor by known malware repositories. Next, we noticed that the same attack patterns were launched repeatedly from different sources worldwide using different attack vectors, all sharing the same attack infrastructure. The Guardicore Global Sensor Network (GGSN) detected the Bondnet in January 2017. The GGSN is a network of deception servers installed in multiple data centers around the world. Based on Guardicore deception technology, it streams early threat information to Guardicore Labs for new attack identification and analysis.

While most victims are used for mining, other victims are used to conduct attacks, serve up malware files or host the C&C servers. The attacker uses the compromised machines to expand the botnet attacking infrastructure, hiding these machines among legitimate servers.

The attacker uses a mix of old vulnerabilities and weak user/password combinations to primarily attack Windows Server machines. The attack vectors we uncovered include known phpMyAdmin configuration bugs, exploits in JBoss, Oracle Web Application Testing Suite, Elasticsearch, MSSQL servers, Apache Tomcat, Oracle Weblogic and other common services.

Common to all these attacks is a series of Visual Basic files that download and install a remote access trojan (RAT) and a cryptocurrency miner. Let's take the MySQL attack vector as an example of how this attack works.

Anatomy of a MySQL Attack

The infection kicks off by planting files using the INTO DUMPFILE command in MySQL and then attempts to load them using MySQL's plugin feature. These commands require unique privileges (among them FILE and EXECUTE) and should be blocked by default. The files are DLLs that execute a payload that runs an obfuscated Visual Basic script. The payload is executed by loading the DLL into the MySQL process and calling the *allcone* exported function. Executing from MySQL's high privileges solves any permission problem the attacker might run into and helps evade antivirus heuristics.

The use of code obfuscation and reliance on old features with weak auditing and security mechanisms allows the attacker to bypass protection measures. In comparison to the newer and more suited to the modern era Powershell, Visual Basic code is hard to log and access to WMI objects is rarely audited. Another advantage to using Visual Basic is its backward compatibility for earlier Windows operating systems (e.g. 2003, 2008).

Visual Basic Payload

The Visual Basic script collects system information such as OS version, number of CPU cores, the current language and network connectivity. Using a criteria based on OS version, number of CPU cores and the OS language (Chinese or not), the script picks what to install. The OS language is used to distinguish between victims in China and elsewhere probably due to communication issues (the [Great Firewall](#)). Most victims receive code that installs a miner and a backdoor while desktop machines inside China receive a browser ad hijacker instead of a miner.

Miner

The miner's installer picks which cryptocurrency to mine and then downloads and installs a matching miner, with priority given to [Monero](#). Additional miners include [ByteCoin](#), [RieCoin](#) or [ZCash](#), all convertible to USD. To acquire the miner, the malware communicates with a hard coded file server picked from the pool of servers used as part of the attack using HTTP over port 4000. The installer prefers to install up to date software, searching for and terminating old copies (if any exist). Last, to make sure the miner process is reboot proof and constantly running, a scheduled task, triggered hourly, activates the miner.

WMI Trojan

The backdoor is a WMI RAT downloaded from an Amazon S3 bucket (named *mytest01234*). The trojan is installed using a known [MOF file method](#) and set to run every night at 11PM. This backdoor works by defining a new WMI provider class inheriting from `__EventConsumer` by the name of *ASEventConsumerdr*. This enables the attacker to execute code as a result of a WMI event such as a specific time period, hiding its activity inside the noise of the WMI service process. Simple yet effective.

The backdoor starts by enabling the Guest account and resetting its password, enabling the attacker to remotely connect to the victim if required using RDP, SMB, [RPC](#) etc.

Then it collects and transmits computer metadata to a C&C server directed from *mst.mymst007.info*.

- Computer name
- RDP Port

- Guest username
- OS Version
- Number of active processors
- Uptime measured in hours
- The original infection vector
- Whether the victim is running a Chinese version of Windows
- OS language
- CPU architecture (x86/x64)

The data is encoded and sent over HTTP as a list of numbers, corresponding to ASCII character values of the data shifted from their original value by two.

Last, the malware downloads a command file from *mst.mymst007.info* over port 8080 and executes it. The commands are obfuscated Visual Basic code, encoded just like the metadata uploaded to the C&C server. After decoding, the malware executes the given code “in memory” using built-in language features, avoiding any writes to disk which might be picked up by logging mechanisms and security software scanning for file activity.

The command we’ve observed downloads another file named *jk.zip* from the C&C server, which is an up to date version of the miner installation code that, if required, replaces the current miner installed on the victim. Oddly enough, this file is saved to disk (as *xw.tmp*) and only then executed, losing the inherent benefits associated with the C&C’s command execution method. Over time we’ve observed several versions of the installation script, all similar to the miner’s installer previously described.

Controlling Bondnet

On each victim-turned-C&C server, the attacker installs a small webserver. This server, a fork of *goup*, is a small open source HTTP server written in Golang. The attacker added to *goup* features such as tracking victims and file encryption on disk using AES. To avoid auditing and firewall alerts, the attacker uses *.asp* and *.zip* file extensions for files served by the C&C server.

The C&C listens on port 8080 and allows the victim to do the following:

- Upload information – Victims can upload information by sending data to *all.asp*. This information is passed in the URL’s query string over a variable named *info*. The server takes this information, encrypts it with a constant AES key and logs it in Base64 format to a file named *za.txt*.
- Download command – Commands are delivered when requesting *alldl.asp*. This file and multiple others (*jk.zip*, *xz.zip*) are saved encrypted on disk and are decrypted in memory before being sent to victims.

This web server has an administration interface allowing common file operations such as uploading files, deleting them, creating folders, etc. All these operations require the client to pass authentication which we were able to bypass, giving us access to the files hosted on all the attacker's C&C and file servers.

Deploying new command and control instances is a manual process. The attacker connects over RDP to the victim machine, copies over a ZIP file (named *ALL-HFS.zip* or *main.zip*) containing the server, an open source service manager (nssm) and command files. Using an included batch script (named *32.bat*), the server is installed with the innocuous name of *w3wp* (same name as the Microsoft IIS host process) and runs on startup. Occasionally the attacker chooses not to create the *all.asp* file and the result is that some C&C servers don't save the information the trojans send.

Attack Infrastructure

The entire infrastructure is built of compromised servers that have been assigned specific roles such as C&C servers, file servers, scanning servers etc. To find new targets, the attacker scans the internet using WinEggDrop, a TCP port scanner available on hacker forums, that provides an updated list of IPs with open ports. Given a list of IP ranges, the scanning server checks for the existence of public unpatched vulnerabilities in different frameworks such as Apache Struts 2 and stores these IPs. Other servers are dedicated attack machines for specific attack vectors such as MSSQL and phpMyAdmin unpatched vulnerabilities. Many victims are repurposed as file servers to serve up mining software to different victims over HTTP on port 4000. The attacker deploys the same web server deployed on the C&C servers on these file servers.

Unlike the rest of the infrastructure which is built of compromised servers, the attacker uses Amazon S3 bucket to host the WMI trojan files.

Who are you Bond007.01?

Bond007.01, the attacker behind Bondnet, also goes by the alias leebond986 that shows up in multiple pieces of code compiled by him, including prior and modern variants of the miner software. The attacker's habit of reusing his own code and having very simple constructs makes us believe the attacker operates alone.

We have multiple reasons to assume that Bond007.01 is based in China:

- The attacker copies and pastes code into his tools from Chinese websites even though non-Chinese websites are equally available to him.
- The attacker's code handles Chinese desktop victims differently compared to all other victims.
- The Bondnet C&C server is compiled on a Chinese computer.

Detection and remediation

To check whether your machine is infected and to clean it, you can EITHER use the Guardicore detection & cleanup script by simply running: *cscript.exe GC-BondnetCleaner.vbs*

OR

do it manually using the following series of steps:

Log files

Every compromised Windows machine hosts at least one of the following log files:

- *%windir%\wb2010kb.log* – Contains a log of a successful attack
- *%windir%\tempdfvt.log* – Contains the log message from running the WMI trojan.

WMI Trojan

To check whether the backdoor is currently installed, run the following command:

```
gwmi -Namespace "root/subscription" -Class __EventConsumer | where name -eq "MYASECdr"
```

If the results include an instance of the *ASEventConsumerdr* it means the trojan is currently active.

To remove it, run the following commands. Note that the first two commands remove the Event callbacks and the third command removes the WMI provider instance; the last two commands remove the class object, preventing an easy return.

```
gwmi -Namespace "root/subscription" -Class __EventFilter | where name -eq "EF" | Remove-WmiObject
```

```
gwmi -Namespace "root/subscription" -Class __EventFilter | where name -eq "EFNMdr" | Remove-WmiObject
```

```
gwmi -Namespace "root/subscription" -Class ASEventConsumerdr | Remove-WmiObject
```

```
$query = "SELECT * FROM meta_class where __class = 'ASEventConsumerdr'"
```

```
Get-WmiObject -Query $query -Namespace "rootsubscription" | where Name -eq "ASEventConsumerdr" | Remove-WmiObject
```

Miner

To locate the miner, check for the existence of a scheduled task with a short name, either “gm”, “ngm” or “cell” that runs a batch script. If detected, remove it along with the miner directory.

You can also filter suspicious tasks with the following two commands

```
SCHTASKS /Query /V /FO LIST /TN gm
```

```
SCHTASKS /Query /V /FO LIST /TN ngm
```

```
SCHTASKS /Query /V /FO LIST /TN cell
```

In some cases the batch script that runs the miner is executed directly by the trojan (without a scheduled task). To kill it, look for one of the miner’s possible process names in the process list.

Backdoor User

If prior steps have shown your machine is compromised, inspect the following as well:

- Guest account: If you are not using your Guest account reset its password and disable it. If you are using it, reset your password.
- Unknown local users: Make sure there are no unknown local users, particularly users under the Administrators security group, for example we’ve observed parts of the botnet adding a user named “webadmin”.
- RDP connections: To complete the cleanup, make sure that RDP connections are disabled unless required. To determine whether RDP connections are disabled, check the registry key using the following command:

```
reg query “HKLMSYSTEMCurrentControlSetControlTerminal Server” /v  
fDenyChildConnections
```

If the value is 0, RDP connections are enabled.

Summary

In this campaign, the attacker’s objective is to mine cryptocurrencies, a task which requires large amounts of CPU/GPU power. This is why our attacker focus on servers, rather than the more easily attacked consumer IoT devices. While the practice of Bitcoin mining has shifted to large commercial vendors (particularly in China), it’s still profitable to privately mine alternative cryptocurrencies. Especially if you’re not paying for power.

Building an attack infrastructure on top of victim machines helps conceal the attacker’s true identity and origin of the attack. It also provides high availability infrastructure, which is very helpful when relying on compromised servers, in case one of the servers fails or loses connectivity to the internet.

While organizations can treat this as an issue of increased electric bills which can annually result in additional costs of 1000-2000\$ per server, this may only be the beginning. With relatively simple modifications the Bondnet can use its complete control over compromised organization servers, many of which contain sensitive information, to spread evil and perform other illegal actions. Today's mining may easily become a ransomware campaign, data exfiltration or lateral movement inside the victim's network.

If you want to make sure your machines are not infected, download our free detection & cleanup tool below.

Written by

[Daniel Goldberg](#)

June 08, 2020

Written by

[Daniel Goldberg](#)

Daniel is a security research expert at Guardicore Labs, where he is responsible for tracking the latest security intelligence, including detailed analysis of hackers' methodologies, for use in creating advanced countermeasures for Guardicore products and services. Daniel has more than 10 years of cybersecurity research experience.