MCCrash: Cross-platform DDoS botnet targets private Minecraft servers | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2022-12-15 · Archived: 2026-04-05 16:35:09 UTC

April 2023 update – Microsoft Threat Intelligence has shifted to a new threat actor naming taxonomy aligned around the theme of weather. **DEV-1028** is now tracked as **Storm-1028**.

To learn about how the new taxonomy represents the origin, unique traits, and impact of threat actors, and to get a complete mapping of threat actor names, read this blog: **Microsoft shifts to a new threat actor naming taxonomy**.

Malware operations continue to rapidly evolve as threat actors add new capabilities to existing botnets, increasingly targeting and recruiting new types of devices. Attackers update malware to target additional operating systems, ranging from PCs to IoT devices, growing their infrastructure rapidly. The Microsoft Defender for IoT research team recently analyzed a cross-platform botnet that originates from malicious software downloads on Windows devices and succeeds in propagating to a variety of Linux-based devices.

The botnet spreads by enumerating default credentials on internet-exposed Secure Shell (SSH)-enabled devices. Because IoT devices are commonly enabled for remote configuration with potentially insecure settings, these devices could be at risk to attacks like this botnet. The botnet's spreading mechanism makes it a unique threat, because while the malware can be removed from the infected source PC, it could persist on unmanaged IoT devices in the network and continue to operate as part of the botnet.

Microsoft tracks this cluster of activity as DEV-1028, a cross-platform botnet that infects Windows devices, Linux devices, and IoT devices. The DEV-1028 botnet is known to launch distributed denial of service (DDoS) attacks against private Minecraft servers.

Our analysis of the DDoS botnet revealed functionalities specifically designed to target private Minecraft Java servers using crafted packets, most likely as a service sold on forums or darknet sites. A breakdown of the systems affected by the botnet over the three months from the time of this analysis also revealed that most of the devices were in Russia:

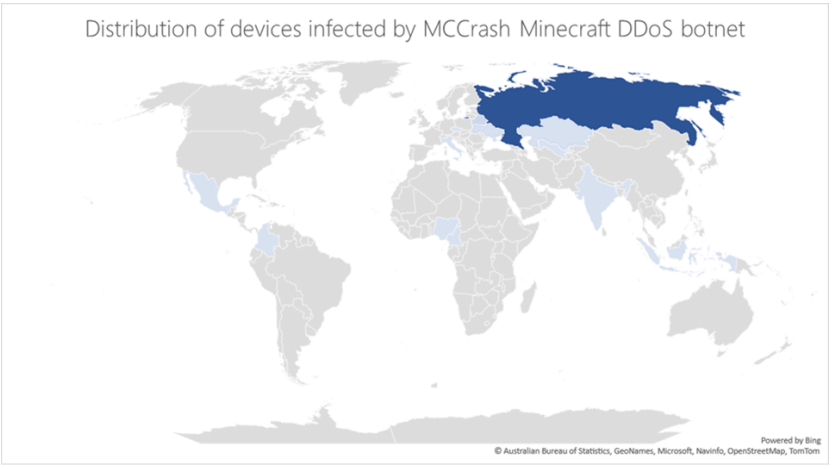

Figure 1. IP distribution of devices infected by the botnet

This type of threat stresses the importance of ensuring that organizations manage, keep up to date, and monitor not just traditional endpoints but also IoT devices that are often less secure. In this blog post, we share details on how this botnet affects multiple platforms, its DDoS capabilities, and recommendations for organizations to prevent their devices from becoming part of a botnet. We also share Minecraft server version information for owners of private servers to update and ensure they are protected from this threat.

Cross-platform botnet targets SSH-enabled devices

Microsoft researchers observed that the initial infection points related to the botnet were devices infected through the installation of malicious cracking tools that purport to acquire illegal Windows licenses.

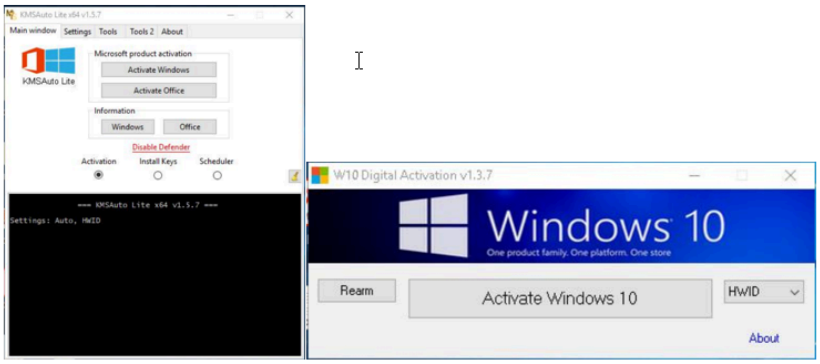

Figure 2. Cracking tools used to spread the botnet.

The cracking tools contain additional code that downloads and launches a fake version of *svchost.exe* through a PowerShell command. In some cases, the downloaded file is named *svchosts.exe*.

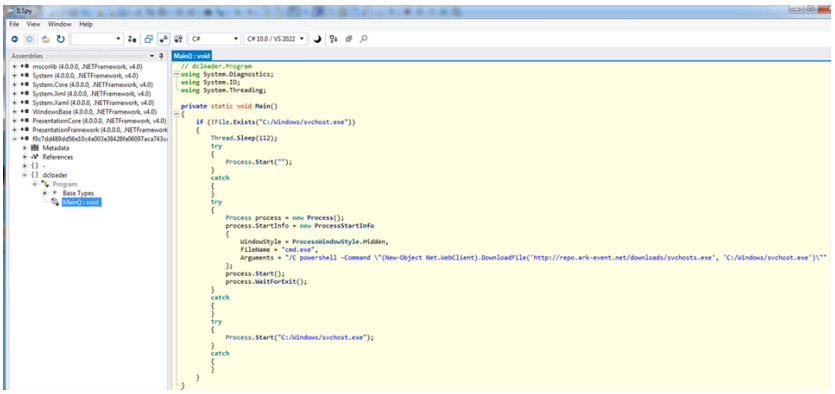

Figure 3. The code of the .NET executable that downloads and runs svchost.exe

Next, *svchost.exe* launches *malicious.py*, the main Python script that contains all the logic of the botnet, whichthen scans the internet for SSH-enabled Linux-based devices (Debian, Ubuntu, CentOS, and IoT workloads such as Raspbian, which are commonly enabled for remote configuration) and launches a dictionary attack to propagate. Once a device is found, it downloads the file *Updater.zip* from *repo[.]ark—event[.]net* onto the device, which creates the file *fuse*. The *fuse* file then downloads a copy of *malicious.py* onto the device. Both *svchost.exe* and *fuse* are compiled using PyInstaller, which bundles all the Python runtime and libraries necessary to initiate *malicious.py*.

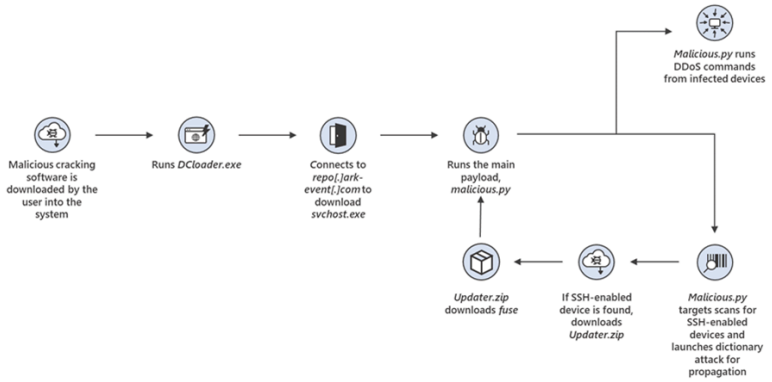

Figure 4. The DDoS botnet attack flow

While *malicious.py* has specific functionalities depending on whether the file launches on a Windows or Linux-based device (for Windows, the file establishes persistency by adding the registry key *Software\Microsoft\Windows\CurrentVersion\Run* with the executable as the value), the executable is compiled to operate on both Windows and Linux-based devices. The file communicates with its command-and-control (C2) server to launch the following commands:

- Establish TCP connection to *repo[.]ark-event[.]net* on port 4676.
- Send initial connection string.
- Receive a key from the server for encryption and decryption, and then encrypt further communication using the Fernet symmetric algorithm.
- Send version information to the server:

    - Windows device: The current Windows version
    - Linux device: Hardcoded version (2.19 in the sample we analyzed)
- Continue receiving encrypted commands from the server

Based on our analysis, the botnet is primarily used to launch DDoS attacks against private Minecraft servers using known server DDoS commands and unique Minecraft commands. Below is the list of commands established in the code:

Command	Description
SYNC	Check that malware is running
PROXY_<url>	Set proxy servers
DOWNLOAD_<url>	Download file
EXEC_<command >	Run specific command line
SCANNER[ON|OFF]	Default credentials attack on SSH servers to spread
ATTACK_TCP	Send random TCP payloads
ATTACK_[HOLD|HANDSHAKE]	Send random TCP payloads through proxy
ATTACK_UDP	Send random UDP payload
ATTACK_VSE	Attack on Valve Source Engine protocol
ATTACK_RAKNET	Attack on RakNet protocol (used by Minecraft servers)
ATTACK_NETTY	Minecraft – Login handshake Packet
ATTACK_[MCBOT|MINE]	Minecraft – Login Start Packet
ATTACK_[MCPING|PING]	Minecraft – Login Success Packet
ATTACK_MCDATA	Minecraft – Login Handshake, Login Start and Close Window Packets
ATTACK_MCCRASH	Minecraft – Login Handshake and Login Start packets, using Username with *env* variable
ATTACK_JUNK	Send Tab-Complete packet
ATTACK_HTTP-GET	Send GET request
ATTACK_HTTP-FAST	Send HEAD request
STOP_ATTACK	Stop the previous attack

While most of the commands are methods of DDoS, the most notable command run by the botnet is *ATTACK_MCCRASH*. The command sends *${env:random payload of specific size:-a}* as the username in order to exhaust the resources of the server and make it crash.

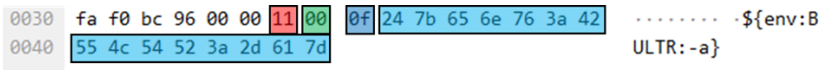

Figure 5. MCCrash TCP payload seen in a packet capture

TCP payloads on port 25565 have the following binary structure:

- Bytes [0:1] – Size of packet
- Bytes [1:2] – Login Start command
- Bytes [2:3] – Size of username
- Bytes [3:18] – Username string

The usage of the *env* variable triggers the use of Log4j 2 library, which causes abnormal consumption of system resources (not related to Log4Shell vulnerability), demonstrating a specific and highly efficient DDoS method.

A wide range of Minecraft server versions could be affected

While testing the impact of the malware, researchers found that the malware itself was hardcoded to target a specific version of Minecraft server, 1.12.2. However, all versions between 1.7.2 and 1.18.2 can be affected by this method of attack. There is a slight modification in the Minecraft protocol in server version 1.19, which was released earlier in 2022, that prevents the

use of the Minecraft specific commands, the *ATTACK_MCCRASH*, *ATTACK_[MCBOT|MINE]* and *ATTACK_MCDATA*, without modification of the attack code.

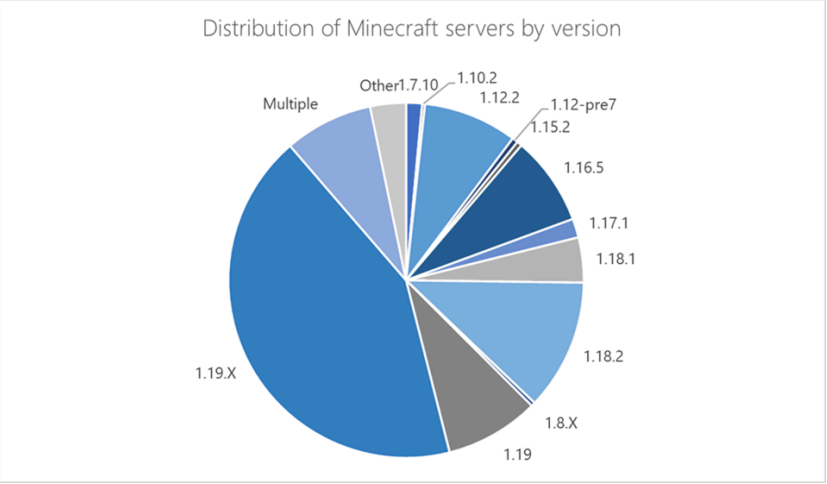

Figure 6. Distribution of Minecraft servers by version

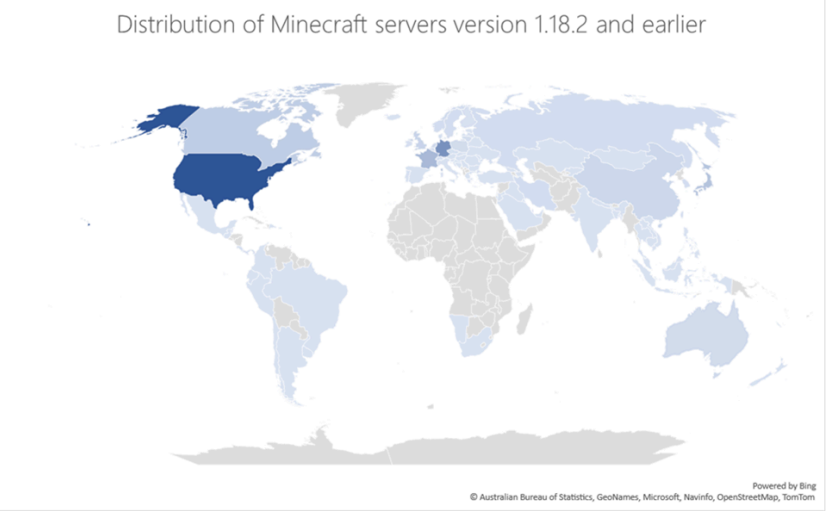

Figure 7. Distribution of Minecraft servers that could be affected by MCCrash

The wide range of at-risk Minecraft servers highlights the impact this malware could have had if it was specifically coded to affect versions beyond 1.12.2. The unique ability of this threat to utilize IoT devices that are often not monitored as part of the botnet substantially increases its impact and reduces its chances of being detected.

Protecting endpoints from cross-platform DDoS botnets like MCCrash

To harden devices networks against threats like MCCrash, organizations must implement the basics to secure identities and their devices, including access limitation. Solutions must detect downloads of malicious programs and malicious attempts to gain access to SSH-enabled devices and generate alerts on anomalous network behavior. Below are some of our recommendations for organizations:

- Ensure employees are not downloading cracking tools as these are abused as an infection source for spreading malware.
- **Increase network security** by enforcing multi-factor authentication (MFA) methods such as Azure Active Directory (now part of Microsoft Entra) MFA. Enable network protection to prevent applications or users from accessing malicious domains and other malicious content on the internet.

  Microsoft 365 Defender protects against attacks related to botnets by coordinating threat data across identities, endpoints, cloud apps, email, and documents. Such cross-domain visibility allows Microsoft 365 Defender to comprehensively detect and remediate end-to-end attack chains—from malicious downloads to its follow-on activities in endpoints. This rich set of tools like advanced hunting let defenders surface threats and gain insights for hardening networks from compromise.

- **Adopt a comprehensive IoT security solution** such as Microsoft Defender for IoT to allow visibility and monitoring of all IoT and OT devices, threat detection and response, and integration with SIEM/SOAR and XDR platforms such as Microsoft Sentinel and Microsoft 365 Defender. Defender for IoT is updated regularly with indicators of compromise (IoCs) from threat research like the example described in this blog, alongside rules to detect malicious activity.

  On the IoT device level:

  - Ensure secure configurations for devices: Change the default password to a strong one, and block SSH from external access.
  - Maintain device health with updates: Make sure devices are up to date with the latest firmware and patches.
  - Use least privileges access: Use a secure virtual private network (VPN) service for remote access and restrict remote access to the device.

- For users hosting private Minecraft servers, update to version 1.19.1 and above.

- **Adopt a comprehensive Windows security solution**
  - Manage the apps your employees can use through Windows Defender Application Control and for unmanaged solutions, enabling Smart App Control.
  - For commercial customers, enable application and browser controls such as Microsoft Defender Application Guard for enhanced protection for Office and Edge.
  - Perform timely cleanup of all unused and stale executables sitting on your organizations' devices.
  - Protect against advanced firmware attacks by enabling memory integrity, Secure Boot, and Trusted Platform Module 2.0, if not enabled by default, which hardens boot using capabilities built into modern CPUs.

Indicators of compromise (IOCs)

- e3361727564b14f5ee19c40f4e8714fab847f41d9782b157ea49cc3963514c25 (KMSAuto++.exe)
- 143614d31bdafc026827e8500bdc254fc1e5d877cb96764bb1bd03afa2de2320 (W10DigitalActivation.exe)
- f9c7dd489dd56e10c4e003e38428fe06097aca743cc878c09bf2bda235c73e30 (dcloader.exe)
- 4e65ec5dee182070e7b59db5bb414e73fe87fd181b3fc95f28fe964bc84d2f1f (updater.zip)
- eb57788fd2451b90d943a6a796ac5e79f0faf7151a62c1d07b744a351dcfa382 (svchost**s**.exe)
- 93738314c07ea370434ac30dad6569c59a9307d8bbde0e6df9be9e2a7438a251 (fuse)
- 202ac3d32871cb3bf91b7c49067bfc935fbc7f0499d357efead1e9f7f5fcb9d1 (malicious.py)
- repo[.]ark-event[.]net

Detections

Microsoft Defender Antivirus

Microsoft Defender Antivirus detects the malware used in this attack as the following:

- TrojanDownloader:MSIL/MCCrash.NZM!MTB
- Trojan:Win32/MCCrash.MA!MTB
- TrojanDownloader:Python/MCCrash!MTB
- Trojan:Python/MCCrash.A
- TrojanDownloader:Linux/MCCrash!MTB
- Trojan:Python/MCCrash.RPB!MTB
- Trojan:Python/MCCrash.RPC!MTB

Microsoft Defender for Endpoint

Microsoft Defender for Endpoint alerts with the following titles can indicate threat activity on your network:

- Emerging threat activity group DEV-1028 detected
- System file masquerade
- Anomaly detected in ASEP registry
- Suspicious process launched using cmd.exe
- Suspicious file launch

Microsoft Defender for IoT

MCCrash-related activity on IoT devices would raise the following alerts in Microsoft Defender for IoT:

- Unauthorized SSH access
- Excessive login attempts

Microsoft Defender for Cloud

Microsoft Defender for Cloud raises the following alert for related activity:

- VM_SuspectDownload

Advanced hunting queries

Microsoft 365 Defender

Run the following queries to search for related files in your environment:

```
DeviceFileEvents
```

```
| where SHA256 in
("e3361727564b14f5ee19c40f4e8714fab847f41d9782b157ea49cc3963514c25","143614d31bdafc026827e8500bdc254fc1e5d877cb96764bb1bd03afa2de2320","f
```

```
DeviceFileEvents
```

```
| where FolderPath endswith @":\windows\svchost.exe"
```

```
DeviceRegistryEvents
```

```
| where RegistryKey contains "CurrentVersion\\Run"
```

```
| where RegistryValueName == "br" or RegistryValueData contains "svchost.exe" or RegistryValueData contains
"svchosts.exe"
```

```
DeviceProcessEvents
```

```
| where FileName in~ ("cmd.exe", "powershell.exe")
```

```
| where ProcessCommandLine has_all ("-command", ".downloadfile(", "windows/svchost.exe")
```

Microsoft Sentinel

Microsoft Sentinel customers can use the TI Mapping analytic to automatically match the malicious domain indicators mentioned in this blog post with data in their workspace. If the TI Map analytics are not currently deployed, customers can install the **Threat Intelligence** solution from the Microsoft Sentinel Content Hub to have the analytics rule deployed in their Sentinel workspace. More details on the Content Hub can be found here: https://learn.microsoft.com/azure/sentinel/sentinel-solutions-deploy

To supplement this indicator matching, customers can use the following queries against data ingested into their workspaces to help find devices with exposed SSH endpoints, and devices that might be under SSH brute force attempts.

Potential SSH brute force attempt: https://github.com/Azure/Azure-Sentinel/blob/master/Detections/Syslog/ssh_potentialBruteForce.yaml

Exposed critical ports in Azure: https://github.com/Azure/Azure-Sentinel/blob/master/Hunting%20Queries/AzureDiagnostics/CriticalPortsOpened.yaml

David Atch, Maayan Shaul, Mae Dotan, Yuval Gordon, *Microsoft Defender for IoT Research Team*

Ross Bevington, *Microsoft Threat Intelligence Center (MSTIC)*
