Kaiten/STD router DDoS Malware

Risk Factor: Medium
Threat Advisory: Kaiten/STD Router DDoS Malware

Editor’s Note: The research published in this advisory predates the very public attacks on www.krebsonsecurity.com in mid-September. The malware examined in this advisory has since been implicated in those attacks. Akamai has chosen to publish this existing research without specific reference to those attacks as the mitigation and cleanup steps are important to publicize. We will be publishing further analysis that includes reference to more recent in the coming weeks.

1.0 / Overview /

Akamai SIRT is investigating a malware variant of Kaiten/STD specifically designed to target networking devices used in Small Office & Home Office (SOHO) as well as DVRs, IP Cameras and other IoT devices. This malware includes an extensive list of available attack vectors along with the ability to execute arbitrary commands and take full control over the infected system.

The malware is packed with a custom packer/encoder to hinder analysis. It’s compiled for multiple architectures (MIPS, ARM, PowerPC, x86, x86_64) and uses a custom IRC-like communication protocol for C2 communications.

In our investigation we have seen the resulting botnets target companies operating in multiple business sectors. Although booter/stresser supported attacks have been more prevalent lately, the largest attacks had previously come from malware-based botnets like XOR and billgates. This variant of malware continues to expand on botnet-based attacks but is also being used in DDoS-For-Hire and extortion campaigns.

2.0 / Indicators of Binary Infection / The infection begins with either an exploitation of a vulnerability or common/guessed/default login credentials. Once the attacker gains access to the target system, a set of predefined commands are run to download and execute ultimately resulting in a malware infection. The malware in turn executes a long list of commands aimed at disabling firewall rules and gain persistence after reboot as shown in Figure 1.
Persistence is configured using the current user's crontab configuration file. It's using a single configuration command to start the malware every minute on the system. The crontab configuration shown in Figure 2.

If we look at the malware's header section, we can see that it's statically compiled and initially includes only 2 sections. The first section with read-and-execute permissions acting as an unpacker stub routine and the second section with read-and-write permissions containing the packed and encoded data. We can use the readelf utility to display the malware's Program Header as shown in Figure 3.
The code in the packer stub routine includes a few syscalls to mmap and mprotect to create an additional section where the new unpacked code will be stored. To manually unpack this sample, we can set a breakpoint to catch all system calls, note the newly mapped memory page and set a hardware breakpoint on execute there. There are no anti-debugging tricks, so finding the Original Entry Point using the above method is straightforward. Once the Original Entry Point has been reached, it is possible to dump the unpacked instructions and continue analysis.

3.0 / Toolkit Analysis / The malware uses a predefined list of C2 IPs and a custom IRC protocol to connect and communicate with them. Once a connection is established to one of the predefined C2 IPs the infected host then authenticates with a dynamic password generated by the server and joins a private channel where it begins listening for commands.

The backdoor functionality of the malware allows the attacker to execute arbitrary commands on the infected machine. These commands include, but are not limited to:

- Download and execute additional malware/scripts
- Set the IP range for spoofed traffic
- Kill/View running processes
- Execute reverse shells
- Resolve internal network domain names
- Execute arbitrary shell commands with the permissions of the current user

The other major functionalities of the malware are aimed at launching DDoS Attacks. The malware presents a variety of attack/flood type options to the botnet operators — including, but not necessarily limited to:

- SYN Flood
- Spoofed/Non-spoofed UDP Flood with custom payloads
- Layer 7 HTTP GET/POST/HEAD Floods
- TCP Connection Floods
- XMAS Flood with customizable TCP Header Flags
4.0 / DDoS Attack Payloads / Many flood types can be generated using this malware. Figures 4 – 10 show the most common attack types and their respective characteristics.

The UDP Flood is generic but allows the control over the payload size and content by the operator as shown in Figures 4 and 5. It can be purposely crafted for bandwidth exhaustion attacks by setting a large payload size or alternatively can be aimed at resource exhaustion with no payload but a high packet-per-second threshold.

![spoofed udp flood with payload](image)

The SYN Flood distinguishes itself by the length of the TCP Headers and their associated options. Figure 6 shows a 40-byte header containing the always present 20-byte TCP Header options.

![tcp header outlining the different tcp header characteristics](image)

The following characteristics are always present at their current offsets:

- TCP Header size of 40 bytes including options
- Max Segment Size of 1460
- Selective Syn-Ack enabled
- NOP at offset 0x38
- Window Scale of x
We have built a Berkeley Packet Filter in Figure 8 that can be used to match some of the traffic generated by the SYN Flood characteristics highlighted in Figure 7.

The TCP XMAS Flood can be set with attacker-controlled TCP flags which can either be run-of-the-mill standard combinations or more infrequently can contain rarely seen abnormal combinations. Sample attack traffic is shown in Figure 9 utilizing these abnormal flag combinations.
the attack to control the name of parameters, create additional GET parameter, change the length of parameters or add a randomized value that will be injected into said parameter. This last parameter’s value will always be numeric.
A noticeable characteristic of the HTTP POST Flood includes an attacker-controlled parameter that appears in GET parameters. These requests don't actually include any POST data as part of the request, even when utilizing the custom parameter and randomized value injection features.

Figure 12: HTTP POST Flood

Figure 13: List of used HTTP User-Agent Headers
Figure 13: List of used HTTP User-Agent Headers (continued)
Figure 13: List of used HTTP User-Agent Headers (continued)
5.0 / **Observed Campaigns** / On June 7, 2016, Akamai’s Security Operations Center detected and mitigated a single attack to one of our clients reaching almost 250 Gbps at its peak. The attack consisted of a UDP Flood with signatures that match the Kaiten/STD malware-generated traffic. Bandwidth statistics are shown in Figure 14, with sample traffic shown in Figure 15.

![Attack Bandwidth Distribution — June 7 DDoS Campaign](image)

**Figure 14: Attack bandwidth per Scrubbing Center**

<table>
<thead>
<tr>
<th>Scrubbing Center</th>
<th>Gbps</th>
<th>Mpps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hong Kong</td>
<td>48</td>
<td>75</td>
</tr>
<tr>
<td>Virginia</td>
<td>26</td>
<td>35</td>
</tr>
<tr>
<td>San Jose</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Frankfurt</td>
<td>72</td>
<td>130</td>
</tr>
<tr>
<td>London</td>
<td>70</td>
<td>80</td>
</tr>
<tr>
<td>Tokyo</td>
<td>22</td>
<td>35</td>
</tr>
</tbody>
</table>

Figure 15: Sample malicious packets taken from DDoS campaign

On April 10, 2016, Akamai detected a layer-7 attack peaking at an estimated 500,000 requests per second. The distribution of this attack's bandwidth is provided in Figure 16.
Figure 16: Attack bandwidth per Scrubbing Center

6.0 / RECOMMENDED DETECTION METHODS / To detect an active infection an admin should check the crontab configuration for all user accounts on the suspect system. An example of an infected system’s crontab is shown in Figure 17.

```
user@Ubuntu:/home/user/Desktop# crontab -l
* * * * * /home/user/Desktop/{malware filename} > /dev/null 2>&1 &
user@Ubuntu:/home/user/Desktop#
```

Figure 17: Crontab showing an active infection

Since the auto-start crontab entry gets inserted only on malware execution, the user can safely remove it and proceed with stopping the existing malicious running process.
The process is identified by the name *irq* and can be found using the *ps* utility as shown in Figure 18.

```
user@Ubuntu:/home/user/Desktop# ps -Fp $(pidof irq)
UID      PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
user    12103  2144 60    69   164   0 05:28 pts/7    04:29:51 irq
user@Ubuntu:/home/user/Desktop#
```

Figure 18: Identifying the malicious process

To identify that the malware sample is from the Kaiten/STD family, we have created the following Yara rule shown in Figure 19. It’s configured to match on the two unique ASCII sentences found in the packed binary as well as the number of sections.

```
import "elf"

rule STD
{
  meta:
    author = "Akamai SIRT"
    description = "Kaiten/STD DDoS malware"

  strings:
    $s0 = "shitteru koto dake"
    $s1 = "nandemo wa shiranai wa yo,"

  condition:
    elf.number_of_sections == 0 and
    elf.number_of_segments == 2 and
    $s0 and $s1
}
```

Figure 19: Yara rule to detect Kaiten/STD malware

**7.0 / Conclusion** / Botnets built on the back of SOHO and IoT devices are steadily growing and can be leveraged in large-scale scanning, compromise systems through use of default names and passwords, and DDoS campaigns.

Some of these systems are easily compromised with publicly available exploits and knowledge. They can also be weaponized using publicly available attack toolkits and malware. These trends and tactics are unlikely to go away and the relative ease of building and renting these botnets will continue to lower the bar even further for attackers.
Owners should be periodically checking for and applying firmware updates. This will help keep long-lived and publicly accessible exploits from working against these devices by patching the flaws that introduced the exploitation opportunity.

It's also recommended that device owners disable unnecessary ports and services that potentially expose additional attack surface and ultimately present an exploitation opportunity. In some cases, these additional services can be leveraged by attackers for reflected DDoS campaigns without needing to actually exploit the machine. Their ability to leverage the device in these types of attacks is merely an underlying flaw with a particular service or protocol such as uPnP, mDNS, TFTP, etc.

Other measures, like changing default usernames and passwords, would appear to be common sense, yet devices with default credentials exist across the internet in many types (Routers, SCADA, IPcam, NAS, etc.) and are traditionally low-hanging fruit targeted by attackers. These shortcomings are usually exploited via large scale scanning campaigns using existing botnets and publicly available lists of known default, common, and weak credential pairs.

In closing, owners of the devices being targeted and exploited must stay vigilant and attempt to keep these systems up to date and hardened against these attacks. Network owners should research what they’re deploying on their networks and take actionable steps to reduce known and unknown potential risks proactively. The basic defenses that protect networks from the known weaknesses of today are often the same as the defenses we need to protect them from the unknown attacks of tomorrow.