A Hammer Lurking In The Shadows

blog.f-secure.com/a-hammer-lurking-in-the-shadows/

March 29, 2019

And then there was ShadowHammer, the supply chain attack on the ASUS Live Update Utility between June and November 2018, which was discovered by Kaspersky earlier this year, and made public a few days ago.

In short, this is how the trojanized Setup.exe works:

1. An executable embedded in the Resources section has been overwritten by the first-stage payload.
2. The program logic has been modified in such a way that instead of installing a software update, it executes a payload implemented as a shellcode.
3. The payload enumerates the MAC addresses on the victim's system, creates MD5 hashes of them and searches these hashes in a large array of hardcoded values.
4. If there is a match, it downloads hxxps://asushotfix.com/logo.jpg or hxxps://asushotfix.com/logo2.jpg, depending on the payload variant. This is meant to be a second-stage x86 shellcode since it will try to execute it within its own process. However, these URLs are not accessible anymore.
5. If no match, create or update a file "idx.ini". (Added in July 2018 – more details below)

If you're more interested in the technical details, our colleagues at Countercept have made an excellent write-up here.

More researchers jumped on this threat and wrote their own analysis as well, such as here and here.

In this post we will focus more on the differences between the variants we have discovered, and how the payload evolved over time. We will also cover some findings about the MAC addresses.

Timeline

1. **June 2018: The beginning.**

   In the first known versions, the embedded executable in the Setup.exe resource section has been partially overwritten by another smaller executable that contains the shellcode.
   The executable is not encrypted, and has a PDB string which is remarkable to say the least:

   ```
   D:\C++\AsusShellCode\Release\AsusShellCode.pdb
   ```

   The number of targeted MAC address hashes was very low. In the earliest sample we found, there were only 18 devices in scope.

   If there is a match, the shellcode will download the file from the following URL and execute it.

   ```
   hxxps://asushotfix.com/logo.jpg
   ```

2. **Early July 2018: Introduction of the INI file.**

   Some interesting functionality was added. If there is NO match with any of the targeted MAC addresses (which will be the case for most devices), the payload will create or update an INI file “idx.ini”.3 different entries are written with a date value corresponding to 7 days/one week later. Example content if it was created today (2019-03-29):

   ```
   [IDX_FILE]
   XXX_IDN=2019-04-05
   XXX_IDE=2019-04-05
   XXX_IDX=2019-04-05
   ```

   The INI file is stored 2 levels up in the directory structure from where setup.exe is stored.

   So if the executable path is

   ```
   C:\Program Files (x86)\ASUS\ASUS Live Update\Temp\6\Setup.exe
   ```

   then the INI file will be dropped as

   ```
   C:\Program Files (x86)\ASUS\ASUS Live Update\idx.ini
   ```

   More MAC hashes were added per iteration, increasing the number to over 200 in a sample compiled on 23 July 2018.

3. **Mid August 2018: Going stealthy.**

   Then there is a hiatus of a few weeks. Most people were enjoying summer at that time, but it looks like these actors spent that period on rewriting a few things to hide their payload better.The malicious payload is now fully encrypted, and has become real shellcode, i.e. not part of an executable image. Consequently, the PDB string is gone, and there is no compilation timestamp anymore, which makes determining the exact date of creation trickier. From here on, we are resorting to the date of first seen.The list of targets grew again, nearly 300 devices now.

4. **Early September 2018: A new URL.**

   A small but interesting change was that the URL changed to

   ```
   hxxps://asushotfix.com/logo2.jpg
   ```

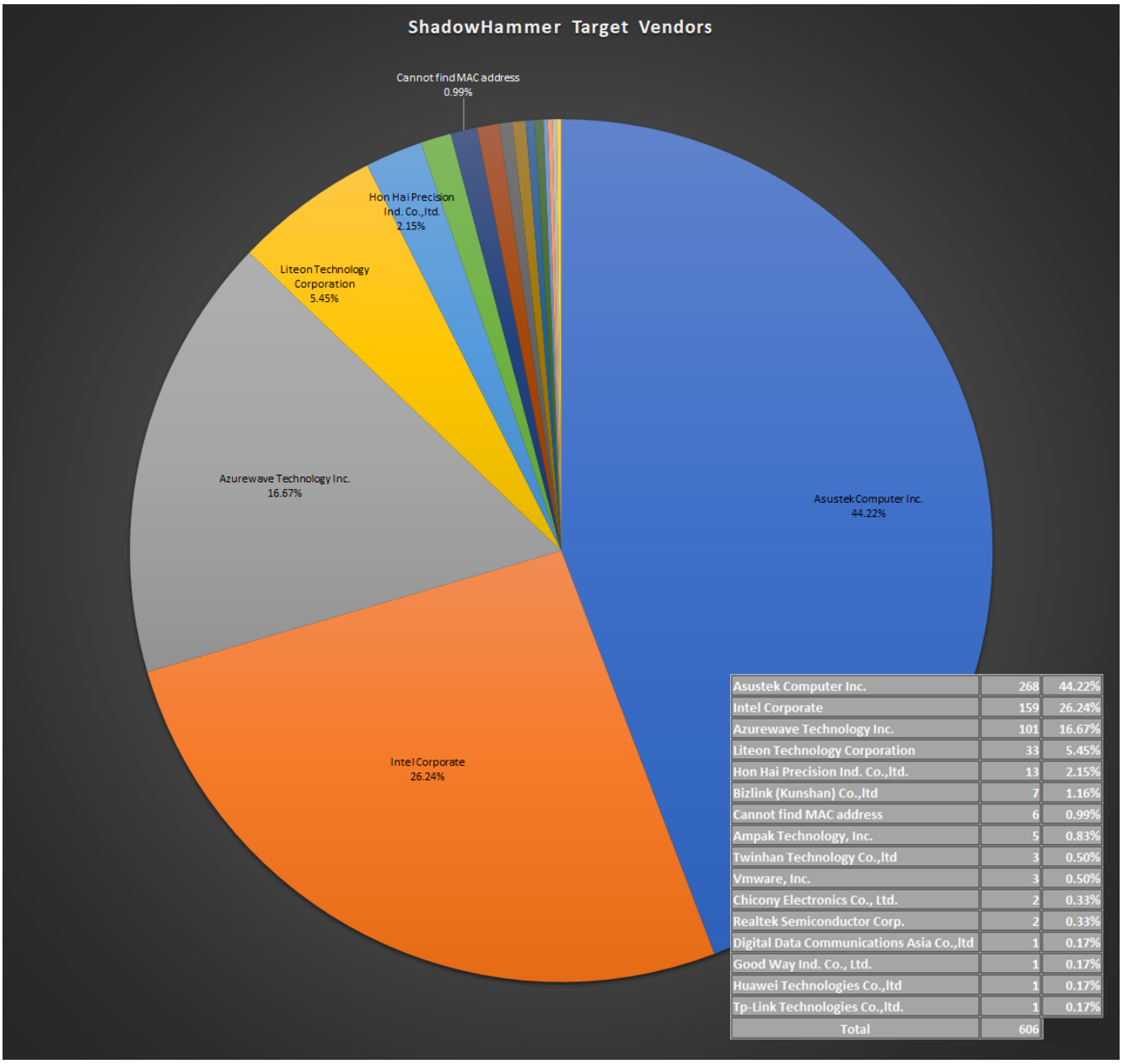
   Also, a few more hashes were added, totaling 307 entries, the largest number we have encountered.

5. **Late September 2018: Revisiting the targets.**

   Until now, the evolution of the targeted MAC addresses was very consistent: the actors have only added targets. In other words, an older sample always contained a subset of the newer sample. Things have changed during the final period of the attack which lasted for more than a month. The number of hashes started fluctuating – with each new variant, some got removed, while new ones were added. Perhaps the threat actors managed to come up with a shortlist of targets of interest this time?

MAC Addresses Observations

Looking at the list of MAC addresses, it appears that some of them are wireless adapters from different manufacturers. It's possible that the attackers gathered these by listening on a wireless network. Also, it suggests that the targets are mostly laptops as most of the wireless adapters seem to be Intel / Azurewave / Liteon.



If you notice in the chart above, there were about 6 MAC addresses that didn't resolve to any vendors:

```
00ff5eXXXXXX
00ff91XXXXXX
00ffaaXXXXXX
00ffd9XXXXXX
0c5b8f279a64
fa94c2XXXXXX
```

0c:5b:8f:27:9a:64, which was found in 8 samples, appears to be a Huawei wireless chip address. It is not assigned to Huawei, but looks like it's being used in Huawei E3372 devices, which is a 4G USB stick. This particular MAC address is always checked along with a specific Asustek Computer Inc. MAC address.

00ff5eXXXXXX is always checked along with a VMWare MAC address, which suggests that this MAC address is used in virtualized environments.

In the most recent sample, there were a total of 18 devices of interest. But here are those that were checked as matches:

- Hon Hai Precision Ind. Co.,ltd. and Vmware, Inc.
- Azurewave Technology Inc. and Asustek Computer Inc.
- Intel Corporate and Asustek Computer Inc.
- Vmware, Inc. and the 00ff5eXXXXXX MAC address

Indicators of Compromise

Hashes

SHA1	DATE (COMPILATION TIMESTAMP/FIRST SEEN)	# OF HARDCODED MAC ADDRESSES	# OF TARGETED DEVICES
b0416f8866954196175d7d9a93b9ab505e96712c	2018-06-12	24	18
5039ff974a81caf331e24eea0f2b33579b00d854	2018-06-28	69	50
e01c1047001206c52c87b8197d772db2a1d3b7b4	2018-07-10	75	55
c6bd8969513b2373eafec9995e31b242753119f2	2018-07-16	156	117
2c591802d8741d6aef1a278b9aca06952f035b8f	2018-07-17	197	152
0595e34841bb3562d2c30a1b22ebf20d31c3be86	2018-07-23	294	208
df4df416c819feb06e4d206ea1ee4c8d07c694ad	2018-08-13	404	287
8e0dfaf40174322396800516b282bf16f62267fa	2018-09-05	433	307
4a8d9a9ca776aaaefd7f6b3ab385dbcfcbf2dfff	2018-09-25	141	86
e793c89ecf7ee1207e79421e137280ae1b377171	2018-09-30	75	41
9f0dbf2ba3b237ff5fd4213b65795595c513e8fa	2018-10-12	22	15
e005c58331eb7db04782fdf9089111979ce1406f	2018-10-19	24	18

YARA Rules

```
// older samples - check the PDB string in the shellcode
rule shadowhammer_pdb
{
    strings:
        $str_pdb = "AsusShellCode.pdb" ascii nocase
    condition:
        all of them
}

// newer samples - check manual patches in the setup.exe

rule shadowhammer_patch
{
    strings:
        $str_msi  = "\\419.msi" ascii wide nocase
        $str_upd = "ASUS Live Updata" ascii wide nocase
        $str_ins = "Asusaller Application" ascii wide nocase
    condition:
        2 of them
}
```