Author: Jialu Wei

Mentor: Ming Chow

Title: DDoS on Internet of Things – a big alarm for the future

Abstract:

Distributed Denial of Service(DDoS) has become a even more popular term on the internet. The attack itself is not any new invention. It is what's behind it that woke people from their dreams: Internet of Things(IoT), now officially and publicly known as the new evil.

As websites and companies start to monitor their servers more diligently, DDoS is less likely to happen directly on any protected/ monitored websites. However, here we are not talking about just computers. The Internet of Things has been expanding rapidly over the past several years. Security cameras, health monitoring systems, your TV, your refrigerator, and probably your coffee maker could be the next security breach. But how do hackers use those small, tangible devices to knock down major websites like Github, Spotify, and Twitter? This paper is going to address how Distributed Denial of Services attacks are conducted through abusing security vulnerabilities in the Internet of Things, as well as what we can do to enhance security of IoT, which is undeniably going to be a big part in the future of technology.

Introduction:

Let us begin with a specific definition of the Internet of Things, commonly known as IoT. The name itself is rather self-explanatory: everything that connects to internet. One definition on the web is that "IoT refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other internet-enabled devices and systems." [i] An important point is that IoT does not just refer to the devices themselves, but the network that all of these devices construct. A network as useful as it is exploitable. Ideally, everything should work perfectly; but in reality, we are putting all the eggs in one basket.

How/why are they vulnerable to attacks:

Let us examine why and how IoT goes wrong. First, IoT is still a relatively new and exciting concept. Even though the idea of IoT was discussed as early as 1982, it was only in 1999 did Bill Joy envisioned Device to Device (D2D) communication, a method that is widely used among the IoT today.[ii] As a result, people are still experiencing the excitement from more and more new inventions, and thus have less time to reflect on what did not work in the older devices.

The second problem might be that the devices are always left on. Imagine the likelihood of you turning off your fridge or your health monitoring system. Almost zero. An IoT device constantly sitting and listening to an open port out there on the internet can be detected by attackers easily with port scanning.

A third concern is related to credential encryption and strength. IoT are usually kept with their default administrative usernames and passwords. One example is the Mirai botnet attack. With 60 of the dumbest default usernames and passwords like "admin" and "1111", Mirai was able to break into 500,000 IoT devices. [iii] Moreover, even when the users change their passwords from default settings, a lot of IoT devices today are storing or sending data with weak encryption or even in plaintext. For example, even trusted companies like Skype have been criticized for allowing unencrypted media in their data path. [iv]

Another reason could be many IoT devices are rarely patched or taken care of. IoT devices were built into households for convenience. However, few people are technically aware enough to go through the trouble to updated their devices. On the other hand, even if some consumers were aware of security issues, in the rush to bring new products and services to market, many companies are likely to overlook long-term support and end up with millions of unpatched computers and mobile devices. Even some major phone vendors stop updating their software in 2-3 year old phones, [v] not to mention that a random $20 IoT device that has been on an house network for years.

How does DDoS happen on IoT:

Distributed Denial of Service attack itself is nothing new. It is an attempt to make an online service unavailable by overwhelming it with traffic from multiple sources, which are usually manipulated. DDoS takes advantage of unprotected computers and manipulates them with malwares to create so-called "botnets". [vi] A botnet is basically a network of "zombie" computers.

Beside password brute-forcing, a 12-year-old vulnerability on OpenSSH also plays a significant role in the creation of botnets. The vulnerability is known as CVE-2004-1653, which is described as "The default configuration for OpenSSH enables AllowTcpForwarding, which could allow remote authenticated users to perform a port bounce, when configured with an anonymous access program such as AnonCVS."[vii] Port forwarding allows remote computers on the internet to connect to a specific device within a private local-area network (LAN). It's an open door to your LAN from the outside. Unfortunately, the TCP port forwarding feature is turned on in many devices by default. It helps attackers to breach the firewalls and hide their tracks by bounding their attacks through any number of IoT that has this vulnerability.

After creating the botnets and gaining remote control over devices, hackers can easily imbed malwares into compromised devices. When the time comes, attackers can initiate a DDoS attack by using their botnets to swarm targeted systems with huge amount of requests and disable the servers.

To the Community:

I chose the topic because I personally experienced much inconvenience during the large scale DDoS attack that knocked down big websites like Twitter and Spotify on October 1st, 2016. I was shocked to learn that the attack was not performed through any fancy new technology, but through many hacked security cameras. It immediately alerted me that while we focus a lot on developing security features on PCs, it is the small things that could knock us out.

So why is this issue important to be discussed? After all, DDoS only concerns the big companies and their benefits. What does that have to do with one's personal life? Truly, DDoS attack itself does not involve too much user privacy issues, like credit card information. However, with the same security breaches and the same mechanisms behind the attacks on big companies, criminals could easily exploit systems in a different way could destroy people's property and privacy. It is only one step away. As this year's holiday season approaches, about 170 million people are expected to buy presents that contribute to the IoT, and research and consulting firm Gartner predicts these networks will grow to encompass 50 billion devices worldwide by 2020. [viii] With this exposure, if we do not discuss the security issues of IoT devices, our money could be stolen, and our houses could be monitored by malicious strangers.

Difficulty with solving it:

As mentioned above, there would be billions of IoT devices expected in the following years. Do we expect all users to configure their own security settings? Sometimes it is just hard to make things secure and easy to use at the same time. As the choices between security, simplicity, and cost can be an intimidating one.

Defense/Action items:

Devices will need more aggressive and solid security features on-board. One of the most obvious feature we can include to battle botnet manipulation is to enforce users to change default passwords and enforce strong passwords. After all, if we do not open the door to the hackers and stop them at the first step, there is less chance that they can embed malwares to manipulate our systems. Also related to personal credentials would be implementing encryption. We would definitely not like someone's credit card information sent in plaintext.

Another important defense a technically informed consumer could take is to shut down unnecessary open ports or services. Breaches like TCP forwarding could be resolved by adding "AllowTcpForwarding No" into the global ssh configuration file.[ix] As for the developers, it is important to develop a proper method to address and securely access any TCP port over the internet without using port forwarding. [x] A better method would even include technology to make ports appear as invisible to random port sniffers.

We should not only focus on the technical defense, but also plant the idea of security into the rapidly growing IoT companies' heads. With all the companies jumping into the industry, it is hard yet necessary to enforce the security standard of a personal computer on any IoT device.

More awareness and investment is the first step. Education should make security a fundamental step rather than an after thought. The right resources, like OWASP (Open Web Application Security Project) with its sections like “IoT vulnerabilities” and “IoT Security Guide”, should be discussed among developers prior to developments.

The madness over the “cool idea” where everything should be connected to the internet might be a problem itself. After all, do we really have to connect everything from hair clips to microwaves? Maybe it is time to encourage sensible consumption. After all, if there is less demand on the market, then perhaps companies wouldn’t be so reckless in pushing out new IoT products. Maybe instead of just having more vulnerable zombie machines, companies could sell fewer products with better quality.

Conclusion:

Internet of Things is undeniably taking over. This article addresses Distributed Denial of Service attack to demonstrate how security breaches of the Internet of Things can be exploited in one way. The scariest part was not the attacks themselves, but what they suggest for the future of IoT. Now we had a DDoS, but what could be next? In our personal lives, there is credit card information. Between companies, there are fund transfers and contracts. Among nations, there is weapon design and transactions. As everything is getting “smarter”, it is important not to let our devices outwit us for some simple, patchable mistakes.

References:

1. Cluley, Graham. "These 60 Dumb Passwords Can Hijack over 500,000 IoT Devices into the Mirai Botnet." *Graham Cluley*. N.p., 10 Oct. 2016. Web. 14 Dec. 2016
2. Dishon, Robin. "DDoS Attacks Explained." *DDoS Attacks Explained*. ESET, 21 Oct. 2016. Web. 13 Dec. 2016.
3. Fadilpasic, Sead. "Hackers Use Old SSH Vulnerability to Attack Internet of Things Devices." *BetaNews*. N.p., 13 Oct. 2016. Web. 14 Dec. 2016.
4. Greenemeier, Larry. "The Internet of Things Is Growing Faster Than the Ability to Defend It." *Scientific American*. N.p., 25 Oct. 2016. Web. 14 Dec. 2016.
5. Hajdarbegovic, Nermin. "Are We Creating An Insecure Internet of Things (IoT)? Security Challenges and Concerns." *Toptal Engineering Blog*. Toptal, n.d. Web. 14 Dec. 2016.
6. Harper, Jon. "Defense Department Moving Slowly on 'Internet of Things' ." *Defense Department Moving Slowly on 'Internet of Things'* N.p., Feb. 2016. Web. 14 Dec. 2016.
7. Kehoe, Patrick. "Is the Internet of Things Too Big to Protect? Not If IoT Applications Are Protected!" *Security Intelligence*. N.p., 22 Aug. 2016. Web. 14 Dec. 2016.
8. Kumar, Mohit. "12-Year-Old SSH Bug Exposes More than 2 Million IoT Devices." *The Hacker News*. N.p., 14 Oct. 2016. Web. 14 Dec. 2016.
9. Paley, Walter. "Securing the Internet of Things | SafeLogic." *SafeLogic*. Safelogic, 02 Jan. 2015. Web. 14 Dec. 2016.
10. Pontin, Jason. "ETC: Bill Joy's Six Webs." *MIT Technology Review*. N.p., 29 Sept. 2005. Web. 14 Dec. 2016
11. "Vulnerability Summary for CVE-2004-1653." *National Cyber Awareness System*. National Vulnerability Database, 31 Aug. 2004. Web. 14 Dec. 2016.

[i] http://www.webopedia.com/TERM/I/internet_of_things.html

[ii] Pontin, Jason. "ETC: Bill Joy's Six Webs." *MIT Technology Review*. N.p., 29 Sept. 2005. Web. 14 Dec. 2016

[iii] Cluley, Graham. "These 60 Dumb Passwords Can Hijack over 500,000 IoT Devices into the Mirai Botnet." *Graham Cluley*. N.p., 10 Oct. 2016. Web. 14 Dec. 2016

[iv] Paley, Walter. "Securing the Internet of Things | SafeLogic." *SafeLogic*. Safelogic, 02 Jan. 2015. Web. 14 Dec. 2016.

[v] Hajdarbegovic, Nermin. "Are We Creating An Insecure Internet of Things (IoT)? Security Challenges and Concerns." *Toptal Engineering Blog*. Toptal, n.d. Web. 14 Dec. 2016.

[vi] Dishon, Robin. "DDoS Attacks Explained." *DDoS Attacks Explained*. ESET, 21 Oct. 2016. Web. 13 Dec. 2016.

[vii] "Vulnerability Summary for CVE-2004-1653." *National Cyber Awareness System*. National Vulnerability Database, 31 Aug. 2004. Web. 14 Dec. 2016.

[viii] Greenemeier, Larry. "The Internet of Things Is Growing Faster Than the Ability to Defend It." *Scientific American*. N.p., 25 Oct. 2016. Web. 14 Dec. 2016.

[ix] Fadilpasic, Sead. "Hackers Use Old SSH Vulnerability to Attack Internet of Things Devices." *BetaNews*. N.p., 13 Oct. 2016. Web. 14 Dec. 2016.

[x] Paley, Walter. "Securing the Internet of Things | SafeLogic." *SafeLogic*. Safelogic, 02 Jan. 2015. Web. 14 Dec. 2016.

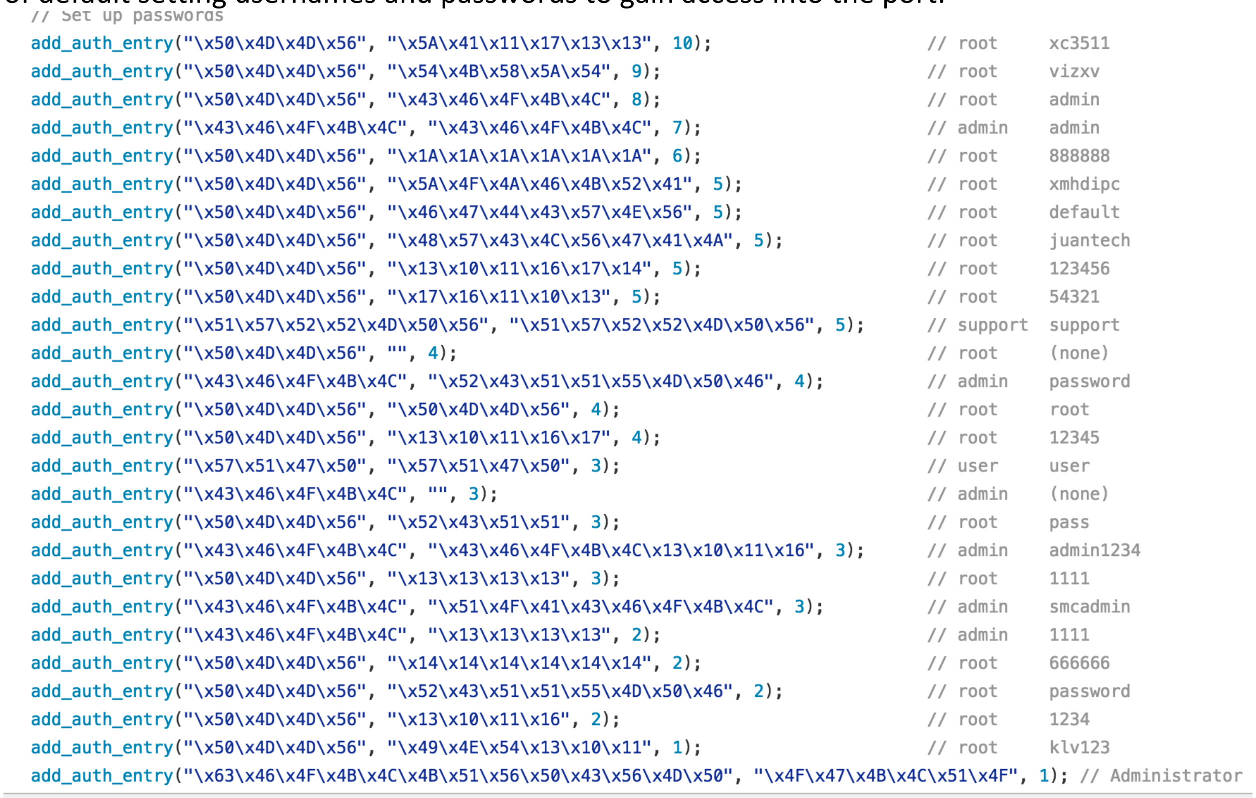
Code Analysis for Mirai – the malware that creates botnet among Internet of Things.
Author: Jialu Wei

I mainly examine the files in the **bot** folder. The folder seems to contain the process of turning a normal computer into a bot, includes killing other processes, scanning for other open ports and services, and the implementation of different attacks. The analysis is done in chronological order (what happens first in the process will by analyzed first):

main.c:
Judging from the comments and debugging outputs, the file seems to set up Command and Control(CNC) sockets to establish the bot's connection with the CNC server. It checks to see if any instance of Mirai was already installed and to make sure the current instance is the only controlling instance. It initialized its "attack", "killer", and "scanner" modules. It also checks for many cases where connection to CNC could fail. It then tries to set up "table".

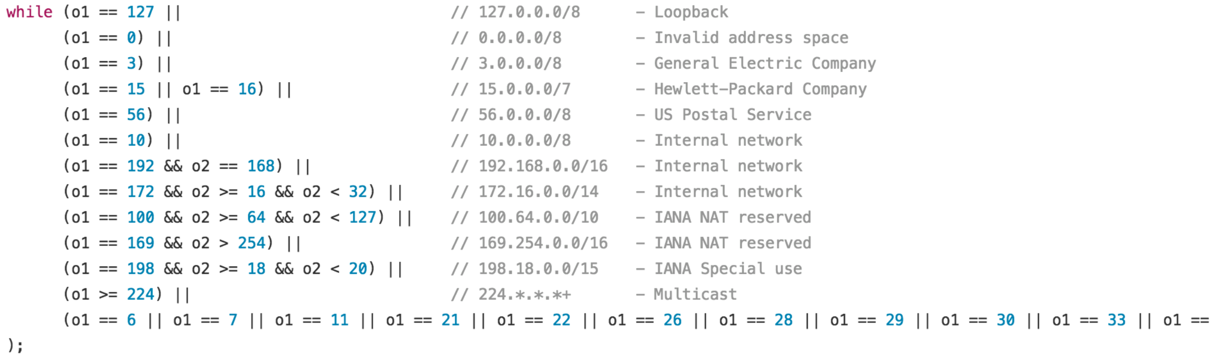
table.h/table.c:
The table here seems to store the configuration of the bot. It has "Generic bot info", the "Killer data", "Scanner data", as well the the "attacking strings". It seems to be a place that stores all the important hard-coded strings and constants.

scanner.c:
It checks open ports by sending SYN request and see the response. If a port is open, it tries a list of default setting usernames and passwords to gain access into the port.



```
// Set up passwords
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);                     // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                          // root     vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);                          // root     admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);                      // admin    admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);                      // root     888888
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x4F\x4A\x46\x4B\x52\x41", 5);                  // root     xmhdipc
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x47\x44\x43\x57\x4E\x56", 5);                  // root     default
add_auth_entry("\x50\x4D\x4D\x56", "\x48\x57\x43\x4C\x56\x47\x41\x4A", 5);              // root     juantech
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17\x14", 5);                      // root     123456
add_auth_entry("\x50\x4D\x4D\x56", "\x17\x16\x11\x10\x13", 5);                          // root     54321
add_auth_entry("\x51\x57\x52\x52\x4D\x50\x56", "\x51\x57\x52\x52\x4D\x50\x56", 5);      // support  support
add_auth_entry("\x50\x4D\x4D\x56", "", 4);                                              // root     (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);          // admin    password
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);                              // root     root
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);                          // root     12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                              // user     user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                                          // admin    (none)
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51", 3);                              // root     pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3);      // admin    admin1234
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x13\x13\x13", 3);                              // root     1111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x51\x4F\x41\x43\x46\x4F\x4B\x4C", 3);          // admin    smcadmin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13", 2);                          // admin    1111
add_auth_entry("\x50\x4D\x4D\x56", "\x14\x14\x14\x14\x14\x14", 2);                      // root     666666
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51\x55\x4D\x50\x46", 2);              // root     password
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16", 2);                              // root     1234
add_auth_entry("\x50\x4D\x4D\x56", "\x49\x4E\x54\x13\x10\x11", 1);                      // root     klv123
add_auth_entry("\x63\x46\x4F\x4B\x4C\x4B\x51\x56\x50\x43\x56\x4D\x50", "\x4F\x47\x4B\x4C\x51\x4F", 1); // Administrator
```

It was funny that in the function “get_random_ip”, the author was trying to avoid the following IPs.



```
while (o1 == 127 ||                             // 127.0.0.0/8      - Loopback
      (o1 == 0) ||                              // 0.0.0.0/8        - Invalid address space
      (o1 == 3) ||                              // 3.0.0.0/8        - General Electric Company
      (o1 == 15 || o1 == 16) ||                 // 15.0.0.0/7       - Hewlett-Packard Company
      (o1 == 56) ||                             // 56.0.0.0/8       - US Postal Service
      (o1 == 10) ||                             // 10.0.0.0/8       - Internal network
      (o1 == 192 && o2 == 168) ||               // 192.168.0.0/16   - Internal network
      (o1 == 172 && o2 >= 16 && o2 < 32) ||     // 172.16.0.0/14    - Internal network
      (o1 == 100 && o2 >= 64 && o2 < 127) ||    // 100.64.0.0/10    - IANA NAT reserved
      (o1 == 169 && o2 > 254) ||                // 169.254.0.0/16   - IANA NAT reserved
      (o1 == 198 && o2 >= 18 && o2 < 20) ||     // 198.18.0.0/15    - IANA Special use
      (o1 >= 224) ||                            // 224.*.*.*+       - Multicast
      (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30 || o1 == 33 || o1 ==
);
```

killer.c:

In the function killer_init, it kills telnet (tcp 23), SSH(tcp 22), and HTTP(tcp 80) services. It also tries to set up exe in certain paths. It could also kill the service on a specific port. It also checks the memory of the computer.

attack.c:

It monitors all the attacks. It adds and conducts the attacks whose implementations are located in the following files:

- attack_app.c: HTTP flood.
- attack_gre.c: GRE ip flood (function attack_gre_ip) and GRE Ethernet attack (function attack_gre_eth).
- attack_tcp.c: SYN flood (function attack_tcp_syn), ACK flood(function attack_tcp_ack), and STOMP protocol attack (function_attack_tcp_stomp).
- Attack_udp.c: UDP flood (function attack_udp_plain) and DNS flood attack (function attack_udp_dns). There are two other functions attack_udp_vse and attack_udp_generic I am not sure what they do.

prompt.txt:

An interesting file I saw in the **mirai** folder. It was written in Russian. It seems like people on the internet are already connecting this attack with Russia. However, my personal opinion is this might just be another spoof. The attribution problem can not be easily solved.