02233 Network Security
Assignments: IoT Security

2024-03-12

In today's lab, we will learn about some of the issues in IoT security. We will start from the attacker perspective studying how IoT botnets work, and walk backward to identify potential vulnerabilities and think of mitigation strategies. In the first exercise, we will investigate the propagation methods for a variant of the Mirai botnet (V3G4) - you can choose another variant if you want, or another IoT botnet altogether -.

DISCLAIMER

Throughout today's exercises, you will be asked to investigate ports and service banners, but please do not conduct any active scanning yourself. Instead, use Shodan, Censys, Greynoise, Virustotal, or other meta-scanners to find relevant information regarding these addresses. While sweep and banner-grabbing scans are legal, they are borderline ethical, and a simple mistake can get you in trouble.

1 Botnets

In this exercise, we will attempt to identify devices compromised by the V3G4 variant of the Mirai botnet (See [2, 3]). This botnet targets IoT devices with known vulnerabilities and/or weak access control (e.g., no authentication, weak credentials or flaws in the authentication method). Start familiarising yourself with how this botnet works to get an understanding of conventional issues in IoT systems connected to the Internet. It is worth mentioning that there are many variants of the Mirai botnet that exploit different vulnerabilities and are run by numerous cyber-criminals [1]; however, the nature of the Mirai botnet is simplistic and less elaborated than other botnets targeting complex systems.

- How does the botnet work?
- What does the botnet do to infect a device?

Solution

The Mirai botnet is known for two main attack vectors: bruteforce attempts (using a small list of credentials), and exploitation of IoT vulnerabilities. The preferred method for expanding is to scan the Internet for open telnet/SSH servers in known ports such as 22, 23, 2222 and 2323. For vulnerability exploitation, the V3GA variant mainly targets Linux environments and embedded devices running Linux embedded.
Moreover, this variant has the C2 domain hardcoded in the Mirai malware, unlike other botnets using Domain Generation Algorithms (DGAs). When a new bot joins the botnet, the Mirai malware terminates processes that close access to the device and removes cohabitating malware. Then, the bot advertises to the C2 server that it has been compromised and is ready to attack.

Services such as Shodan, and GreyNoise act as search engines for Internet-connected devices. Shodan scans the Internet relatively often to find exposed systems, while GreyNoise collects and analyses data from other systems scanning or attacking the Internet. *(Note: The free plan of these services limits the number of results you can get, but you can still use it for this exercise.).*

- Use GreyNoise to find devices matching the attack patterns from V3G4. You can filter the results by CVEs and other tags (e.g.,"**iot tags:Mirai**").
- Use Shodan to find out more about the list of IPs you gathered.
- Which devices did you find? Can you see any common attributes (e.g. open ports, operative system, manufacturer, etc)?
- What can we do to harden the security of your IoT devices? how can we prevent our devices from joining a Mirai botnet?

Solution

Aggregating data from GreyNoise and Shodan gives very useful results with few false positives. Information such as OS, attempted attack, CVE, and more, help us understand whether these devices have been compromised and are malicious. Furthermore, other services such as VirusTotal can enrich our results even further, showing relationship graphs, and how ISPs see these IP addresses. Our level of confidence to say whether a device has been compromised will raise along with the information we gather. For example, we can be almost certain that an IP camera has been compromised if it is running a deprecated version of embedded Linux OS and is attempting to brute force other services or dropping malicious payloads. Whether this information is enough will depend on the circumstances and the context of the device. Therefore, it is important to limit ourselves to the evidence we can gather and present our results accordingly.

To notify consumers we can find publicly available information such as `WHOIS`, contact details (some companies include this information in the banners!), registrars, etc. If the owner is a private entity, you can contact the ISP owner of the IP range. Lastly, some companies have bounty programs that specify how to present valid results. Most bounty programs will give you the chance to investigate further, which can be a great experience or even a carrier path.

***Trivia**: Some bounty programs do not allow active scanning of their network or run aggressive penetration testing tools. On the other hand, Shodan and other meta-scanners are still allowed.*

Botnet investigation can lead to command and control (C2) takeovers and takedowns. In some cases, this can be done by acquiring malware samples from compromised hosts and analyzing the samples to find the C2 servers. Researchers use tools such as VirusTotal (Enterprise) to find malware samples, Ghidra for reverse engineering, Cuckoo to analyze the behavior of the malware in sandboxed environments, and fake infected machines to study the communications with the C2 servers.

If you are curious to test how a botnet works in practice, you can try BYOB, which is a small project that allows you to setup a C2 server and install a bot agent on another instance (e.g., VM or another computer).

2 Exposed IoT devices

Now that we understand how some IoT botnets work, it is time to find other factors increasing the attack surface in IoT devices exposed to the Internet at DTU. For this, we will focus on DTU-only domains (there are many of them),

and try to identify which services DTU exposes to the Internet. There are many departments at DTU, most of them running experiments that require hosting databases, domains, and other services. However, network complexity is typically a synonym for attack surface. Therefore, we are curious to know which devices DTU is exposing to the Internet, what vulnerabilities they have, and how could adversaries take advantage of this situation.

- First, find out which addresses DTU exposes to the Internet. You can use `WHOIS` services, DNS records, or even Shodan to find information about a domain.

  ***Hint**: try "dtu.dk"*

  Solution

  We can go about this problem in many different ways. Our choice is to find out the ISP and ASN responsible for DTU domains through Shodan directly. The ISP is the Danish network for Research and Education (Forskningsnettet), and the ASN is AS1835. This should be enough to find the almost 739 addresses exposed from DTU (for comparison, KU exposes 3000).

- Now, query Shodan or Censys to see which services they expose and their banner information. A banner is the first response a service returns when a client tries to connect to it. You do not need to do this part yourself, you can use Shodan or another meta-scanner to retrieve this information. Focus on IoT protocols, such as the ones botnets target the most (e.g., SSH, MQTT, Telnet, FTP, MySQL...).

  ***Hint**: try the organisation name*

  Solution

  Focusing on IoT protocols, we see a couple of interesting addresses. At this time, 22 of them have SSH ports exposed to the Internet directly, 15 have an FTP server, and 1 of them uses a very old version of MQTT. You can see the services for each port here or IANA associated ones.

- Banners contain many interesting details about the exposed service (e.g., encryption mechanisms, authentication policies, service version, or even OS and other device-specific information). Sometimes, it is sufficient to say that we can retrieve their banner to say the leak sensitive information. Why do you think this is the case? Try to reason the following questions:

  - From the banners you see, which information could be used to gain additional insights into the device's security?
  - Is any of these services leaking sensitive information?

– Could an attacker use the information from these banners to gain an initial foothold into the network?
– How would you mitigate these threats?

Solution

Note that one compromised device can lead to a cascading effect, where other devices that were properly hidden from the Internet before, now are part of the attack surface. Now, if we focus on certain protocols that tend to disclose the most information, and tend to collect vulnerabilities due to the level of access they provide (e.g., SSH and FTP give a shell, MySQL gives the data directly, Telnet sometimes gives a shell as well...) we can see that there are many SSH services with different parameters.
This may not seem obvious at first, but if we use the SSH version as our indicator and query one of the CVE databases (e.g., NVD), we can see that there is a bunch of critical vulnerabilities for SSH services with versions below $v7.2$ (e.g., CVE-2016-1908, CVE-2015-5600, CVE-2016-10009...). These vulnerabilities are not new, some of which are almost 10 years old, which also tell us about the security position of the owner, and perhaps the posture of DTU. It is important to mention that these networks are likely very segmented and separate from others, but this is still a security risk. We could use other indications as well, such as the encryption algorithm used for the key (use weak cryptos?), the key length (maybe too short?), or even the hashing algorithms.
To mitigate this threat, we could notify DTU of this server, the device owner, or perhaps even the ISP. There are loads of research including best practices on how to do this, for example, we could send an email with the details of the vulnerability, some estimation of the risk from our point of view, and some recommendations on how to solve it. Our recommendation for a vulnerable SSH service would be to update the SSH service and remove it from the Internet. Clients should use VPN connections to access servers at DTU whenever possible.

References

[1] MalwareBazaar. *Malwarebazaar — statistics*. URL: `https://bazaar.abuse.ch/statistics`.

[2] Palo Alto Networks. *Mirai variant v3g4 targets iot devices*. URL: `https://unit42.paloaltonetworks.com/mirai-variant-v3g4/`.

[3] Security Week. *Mirai variant v3g4 targets 13 vulnerabilities to infect iot devices - securityweek*. URL: https://www.securityweek.com/mirai-variant-v3g4-targets-13-vulnerabilities-to-infect-iot-devices/.