Action classification and deviation detection of network devices

Zhen XIAO

***Abstract*—This paper was originally submitted to Xinova as a response to a Request for Invention (RFI) on cyber-attack forecasting, detection, and stabilization methods. In this paper, action classification and deviation detection of network devices is proposed.**

In more detail, this describes a solution that performs traffic analysis and detects device anomalies. IoT devices generally have fixed activity patterns, however, personal devices tend to have diverse actions. By monitoring such patterns, and creating better accuracy of device classification, it is possible to more quickly neutralize an attack due to stricter detection criteria for a variety of devices.

I. Introduction

NETWORK devices, such as the growing category of IoT devices, may be hacked and used as origin of attacks during DDoS attacks. Current IDS are mostly based on static rules, which utilize white lists and black lists to perform some level of security. They often use a fixed set of features to match against the ongoing traffic to detect and ban hacked devices. Because these rules are relatively static, they need to be frequently updated, and a hacker can engineer the attacking devise's behavior to circumvent any of those fixed rules.

II. Summary of the invention

The solution consists of a Network Gateway that can classify its devices automatically based on its network traffic pattern, and limit their permitted action set based on its class. When a device's class changes abruptly, we can assume such device is hacked, and can take mitigation actions. An automated, machine-learning based classifier can instead distinguish "regular" traffic, human-generated irregular traffic, and attack traffic solely based on its observation, thus can sustain change of device behavior due to software updates, and the change of attack behavior due to hacker's latest innovation.

III. Detailed Explanation

A network gateway can observe traffic between devices within the network and destination on the internet. Grouping these traffic records based on device, it can obtain a traffic profile of a certain device. The high-level idea of this invention is to ban or limit the traffic of any device whenever its traffic pattern deviates from normal, in order to limit an attack originated from within the network to a destination outside the network (or vice versa). We hereby discuss the necessary design factors when implementing this automated detection mechanism.

Background

A network device produces traffic that includes many features to analyze. For example, by looking at its DNS query packets we can efficiently know the destination it talks to, and by looking at the frequency it produces Network Time Protocol packets to perform time synchronization we can determine if the device has a clock and if the clock is not accurate. (Many IoT devices operate based on a time schedule but does not have accurate hardware clocks).

Part 1. The rigidity of traffic pattern

Some network devices produce a regular, predictable traffic pattern. For example, a thermostat queries the weather service twice a day, or an IoT security camera uploads its video feed constantly to a particular remote endpoint. We say these devices have a highly rigid traffic pattern. Other network devices such as a smartphone connected to Wi-Fi network will produce highly variable network traffic activity, depending on its human user's action.

Based on past traffic pattern, it is very easy to observe whether a network device has rigid traffic pattern. The gateway can then assign a rigidity score for each device.

This device such as POS machines; a computer workstation used to complete fixed tasks, hence running the same application and connect to the same website / service every day, will also have very rigid traffic pattern.

Part 2. Attack detection and mitigation

By running machine-learning clustering algorithm on the traffic pattern, the gateway can then assign devices into different classes: human-interacting devices, and IoT devices. IoT devices can be further categorized into different fixed-function IoT device classes (such as smart-home related, security monitoring related, kids smart-toys, etc.), and general-purpose IoT devices (such as voice assistant). Each device also gets assigned a rigidity score. The algorithm may utilize embedded prior-knowledge from analyzing traffic traces from many different network devices. The algorithm may also use unsupervised learning methods (such as autoencoder) to discover new, unknown classes and summarize their characteristics, which is more future-proof and can adapt to future unknown patterns.

A recurrent-learning machine learning model will see through every packet produced by a particular network device, and for each new packet, produce a score to denote "whether

this packet belongs to the same pattern as all the previous packets' traffic pattern". The frequency and bandwidth information of new traffic can also be feed to the network as its input. The overall score of the new traffic pattern can be the average of all its individual packet.

When a network device starts to produce traffic that does not belongs to its own traffic pattern, the gateway can raise a suspicious flag and scrutiny the anomaly traffic. The gateway can check the dissimilarity of anomaly traffic and previous traffic pattern of the device and compare against the rigidity score to determine if such traffic is likely to be an attack, and assign a confidence score. When the anomaly traffic clearly belongs to a different device class than its originating device, we can immediately conclude such traffic is an attack (with highest confidence score).

After detecting an attack, the gateway has several options.

• When the confidence score is very high, notifying a human administrator is preferred. (Low-score detections may be frequent and there's no need to alert a human every time.) The anomaly traffic can be sample and recorded for future analysis by administrator.

• The gateway may start to block the anomaly traffic, while still allowing that device to send and receive its usual traffic, determined based on its device classification and previous traffic pattern. Conventional IDS may decide to block this device entirely, causing much collateral damage when the attack is a false alarm.

• A device with low-rigidity traffic pattern may produce low-confidence attack alert. In this case, the gateway can throttle the device's traffic to a very low bandwidth; for IoT devices, this typically will not affect its regular functionality.

• It is also possible to only throttle traffic that deviates from each device's usual traffic pattern. In this way, the usual workload on that device will not be affected; using the previous example, that computer workstation can still perform its duty while running the regular applications, using full network bandwidth, while a hacker's attempt to transfer files out of the network will be severely throttled.

In this way, the system can reduce the workload of human operators by automatically limit suspicious traffic, while also minimizing the impact of regular traffic. In conventional system, either alerts are generated without limiting traffic, causing the attack to continue while operators are overwhelmed by false alerts; or alerts will cause devices being completely blocked, causing users affected by false alert requiring administrator to unblock, also generates overwhelming workload.

Part 3. Adapting to device behavior changes

After a device change its "regular" traffic pattern, it may be falsely classified as an anomaly. After generating alert for the user directly, the user of human-interacting devices can manually resolve the issue after, for example, a system update introduces new feature to its general-purpose IoT device or installed new application on personal computer. Here we focus on automatically resolving issue after autonomous IoT device change its behavior.

We hereby assume a special-purpose IoT device will not receive a magic software update and suddenly become a general-purpose device, or become a special-purpose device in another completely different device class.

After legitimate software update, the IoT device may change from one rigid traffic pattern to another rigid traffic pattern; for example, from connecting to Yahoo Weather API to connecting to Amazon Cloud Service, due to Yahoo closing the API. Since the two pattern is different, the gateway's anomaly detection will initially determine the new traffic as anomaly, with high confidence. Meanwhile, the traffic pattern classifier will recognize that the new traffic is also highly rigid and belong to the same class of special-purpose IoT device, thus less likely to be an attack. After being throttled for a period of time, the device will be removed from quarantine and recover full access to network.

Part 4. Synchronizing the knowledge

Different network gateway devices belonging to the same "security colloquium" can share their knowledge and findings regarding the detection of anomaly traffic and exchange their confidence. When many of them discover the same anomalous traffic appears on a specific type of IoT device, it is highly likely that those devices are experiencing epidemic attack, and other gateways can take precaution action against that type of devices in their network. The colloquium of devices may also automatically alert the manufacturer of such IoT device, or provide intelligence for network security personnel. Meanwhile, a software update for IoT devices can also be pre-announced within such colloquium by the manufacturer.
As an example, all Cisco IDS boxes and Cisco home modems can form such a colloquium.

Fig. 1. IoT devices becoming ubiquitous, increasing danger.

Outlier detection can be done using any desired characteristics, for example by using K-nearest-neighbor grouping on parametrically reduced representation. In practice, network data will have many dimensions to analyze.

IV. Conclusion

This solution can possibly be made available as firmware update to existing network gateways, thus can expect wide adoption within 2~4 years. The merits of this implementation would be minimal harassment for human users, in a mixed IoT-human network. There would be faster neutralization for attack, due to stricter detection criteria for IoT devices. This could lead to better accuracy of classification by collaboration between different networks.

The algorithm can already run on existing IDS middleboxes. However, its training requires collecting traffic traces from a variety of devices in normal settings, with as many different types of devices as possible.

Acknowledgment

This research was originally submitted to Xinova, LLC by the author in response to a Request for Invention. It is among several submissions that Xinova has chosen to make available to the wider community. The author wishes to thank Xinova, LLC for their funding support of this research.

More information about Xinova, LLC is available at www.xinova.com.

References

[i] https://blog.trendmicro.com/trendlabs-security-intelligence/spam-remote-access-trojan-adwind-jrat/ community

[ii] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis," IEEE Access, vol. 3, pp. 1132–1142, 2015.