Vol.2018-IOT-43 No.3
Vol.2018-SPT-30 No.3
2018/9/27

What Inventory Should be Prepared for a Security Incident and How?

Motoyuki OHMORI[1,a)]

Abstract: Computer security has been getting more attentions because a computer security incident may cause great damage on an organization. It may be then said that an inventory of hosts should be necessary in order to prepare for a security incident. It is, however, difficult to build up an inventory and keep it up-to-date. In addition, it is still unclear which items of an inventory of hosts are really necessary and which items are unnecessary. This paper then discusses the necessary items for an inventory from the viewpoint of feasibility. This paper also discusses how an inventory should be maintained regarding a security incident. This paper then presents how to automatically collect inventory items in order to make it easy to maintain an inventory.

1. Introduction

Computer security has been getting more attentions because a computer security incident may cause great damage on an organization. An incident is usually detected by Intrusion Prevention System (IPS), Intrusion Detection System (IPS) or Security Operation Center (SOC) service. An incident is then identified by an IP address of a suspicious host, an IP address of corresponding malicious hosts, port numbers and so on. In order to respond against an incident, it is necessary to identify a suspicious host and its user. Some sort of an inventory of all hosts in an organization network should be then maintained. It is, however, difficult to keep an inventory up-to-date. In addition, it is unclear what information is necessary for an incident response.

This paper then proposes to define required items for an inventory such as an IP address, MAC address, Point of Contact (PoC), a connected port on a switch and so on. This paper also proposes how to maintain an inventory for an incident, i.e., ARP table polling is not accurate enough and an inventory should be automatically maintained.

The rest of this paper is organized as follows. **Sec. 2** presents our motivations to define required items for an inventory, and presents that manual maintenance are not accurate and traditional ARP table polling is not accurate. **Sec. 3** tries to define required items for an inventory for an incident. **Sec. 4** proposes automated inventory item collecting. **Sec. 5** refers to related work. **Sec. 6** finally concludes this paper.

2. Background

This section presents a background why we try to define required items for an inventory for an incident and why to automate an inventory collection. This section also presents why traditional ARP table polling is not accurate.

Not all members of even Computer Security Incident Response Team (CSIRT) understand what information is necessary for an incident response. For example, some of CSIRT members do not realize a possibility that an IP address is assigned to a suspicious host without any authorization. To make matters worse, an IP address authorized to a host may be assigned to other host by malicious software. We have, indeed, experienced that unknown equipment made by Apple, inc. assigns unauthorized IP address to its own NIC and frequently changes the IP address. In order to identify such equipment, we need to know what host is connected to which port on which switch, i.e., a port list of a switch. Some of CSIRT members, however, do not realize such possibility, and it does not occur to them that a port list of a switch must be maintained. We, Tottori University, then have no port lists of all switches. It may be then necessary to define required items for an inventory for an incident.

Let us assume that port lists are maintained somehow. The port lists might be usually useful while the ports might not be useful in some situations. For example, we, Tottori University, have a switch that has a unused but configured access port. Our members might connect an unauthorized host to the access port without any authorization. We have already seen such situations. It is then difficult to keep an inventory to be up-to-date. It is almost impossible to manually maintain an inventory of hosts. An inventory should be automatically maintained.

We have been collection bindings of IP addresses and associated MAC address at core switches by polling MAC address tables. The polling interval may usually be 5 min. We have, however, experienced that unknown equipment made by Apple, inc. assigns unauthorized IP address to its own NIC as described above. In addition, such unknown equipment frequently changes the IP address. There actually was a security event that seemed to be caused by such unauthorized equipment, and it was reported

[1] Tottori University, Koyama-minami, Tottori Japan, 680–8550 Japan
[a)] ohmori@tottori-u.ac.jp

© 2018 Information Processing Society of Japan

by National Institute of Informatics Security Operation Collaboration Services, the so-called NII-SOCS, operated by National Institute of Informatics (NII) [1]. We could not find a MAC address of such equipment in our ARP table records polled at the interval of 5 min. It can be said that 5 min. polling of ARP table is then not accurate enough for an incident response. We then need some sort of methods to track a binding of an IP address and MAC address more accurately.

3. Inventory for Incident

This section defines required items for an inventory for an incident.

We here prioritize feasibility to implement an inventory and maintain in an actual operation. A critical incident that has a possibility to incur data breach rarely happens. We then decide not to collect an item in an inventory that requires heavy operation such as confirmation to a user and that is required when and only when a critical incident happens. As shown in **Table 1**, we, Tottori University, now actually have the small number of critical incidents. We then decide to collect an item in an inventory that can be easily or automatically collected. **Table 2** shows items required in an inventory. Ones may consider that an inventory includes confidential information existence on a host because such information is useful on an incident. It is, however, difficult to confirm its existence and maintain.

4. Automated Inventory Collection

This section presents how to automatically collect inventory items. This section firstly presents how to automatically collect a binding of an IP address and MAC address. This section then presents how to automatically collect a location of a host, i.e., what host is connected to which port on which switch. This section then presents how to automatically identify an actual user of a host.

4.1 IP Address and MAC Address Bindings

ARP table polling is a traditional way to collect IP address and MAC address bindings. It is, however, not accurate enough as described in **Sec. 2**. We here propose an efficient passive method to collect ARP table entries. Our proposal captures traffic at core switches, parses ARP traffic, and then builds up ARP table entries. It may be a heavy load to parse all traffic forwarded by core switches. We then decide to utilize *policy-based mirroring* implemented on AlaxalA core switch, AX8600 series. *Policy-based mirroring* enables to capture specific traffic only, e.g., ARP frames only.

Our proposal then records a time when a MAC address associated with an IP address is resolved. Our proposal can then record multiple MAC addresses for an IP address at the same time, and can work well even when the same IP address is duplicated assigned to multiple hosts.

4.2 Host Locating Using Network Authentication

Host location here implies what host is connected to which port on which switch. In case of wireless LAN, IEEE802.1x authentication becomes common, and it enables to locate a host. We then focus on not wireless LAN but wired LAN.

In case of wired LAN, IEEE802.1x authentication is still uncommon. For example, Windows OS before Windows 10 1709 does not enable IEEE802.1x authentication on wired LAN NIC. A user should have enabled a special service by himself/herself, and its configuration is not so easy. In addition, it is very difficult for a user to authenticate a RADIUS server of IEEE802.1x authentication. More specifically, a user must specify a specific root CA for IEEE802.1x authentication only because trust anchors of PKI are not shared within OS. In case of macOS, it is also difficult to specify a RADIUS server, and it requires special configuration. In addition, there are some switches that do not forward and drop IEEE802.1x frames. If a user installs such switch, a host of a user cannot be authenticated. It is then difficult to deploy IEEE802.1x authentication in a whole network in an organization.

We then simultaneously implemented IEEE802.1x authentication, MAC address authentication and Web authentication. A host can be connected to a network if at least one of these authentication succeeds. It may be difficult to suddenly deploy a network authentication in some places. In such places, we configured a RADIUS server to accept all MAC address on MAC address authentication. This enables even unauthorized host to connect to a network. This, however, records which MAC address is seen on which port of which switch. Locations of all MAC addresses of hosts in a network can be then recorded. In order to make it easier and faster to track a host location on an incident, we utilized *line log* of FreeRadius instead of default log. We then configured that RADIUS servers sent *line log* via syslog. We then stored the latest 2-month logs in mongoDB via fluentd. We also utilized a RADIUS accounting that clarified when a host starts to connect to a network and when finishes.

4.3 User Identification

User identification is necessary in a network where network authentication except for *all-accepting MAC address authentication* is not implemented. In order to identify a user of a host in a such network, we utilize authentication logs of Shibboleth IdP, Mail Retrieval Agent (MRA), and groupware. These logs are forwarded using syslog, and they are also stored into mongoDB. MongoDB holds 2-months logs as same as authentication log. MongoDB holds following fields for user identification:

- an IP address,
- a user ID, and
- login time.

5. Related Work

Information Security Management System (ISMS) ISO/IEC-27001[2] briefly defines requirements of computer security incident responses. There are many security or network vendors such as TrendMicro, Paloalto, FireEye, Fortigate, Cisco, Alaxala and so on try to produce the best security solutions. NAGAI, Y. et al. investigated and reported differences between ISMSs in national universities in Japan [3].

6. Concluding Remarks

This paper has presented the minimal items for an inventory for

Table 1 Summary of security events and critical incidents in Tottori University.

fiscal year	the number of events	the number of incidents	the number of critical incidents	remarks
2014	N/A	7	6	
2015	N/A	10	9	
2016	N/A	38	14	Since July, manually remove malware attached to mails. Since August, remove executable files attached to mails. Since the end of August, remove executable files attached to mails. Since October, quarantine suspicious mails.
2017	68	30	1	Join NII-SOCS. On September, install next generation firewall.
2018	20	4	0	as of 30th August 2018.

Table 2 Required host inventory items for an incident.

item	remarks
an IP address of a host	
a MAC address of a host	
a user	an actual user of a host.
a PoC	a person in charge of an incident response.
a connected port	
a connected switch	
observed times	all observed times should be recorded.

a computer security incident that might be feasible to maintain. This paper has also proposed to automatically collect inventory items. An implementation is future work.

References

[1] National Institute of Informatics: National Institute of Informatics, `http://www.nii.ac.jp/` (2007). Accessed: 2017/05/26.

[2] ISO/IEC: Information Security Management Systems Requirements (2013). ISO/IEC27001:2013.

[3] NAGAI, Y., TADAMURA, K. and OGAWARA, K.: Considering Incident Management Systems in Some National Universities, *SIG Technical Reports*, Vol. 2014-IS-127, No. 7, pp. 1–7 (2014).

© 2018 Information Processing Society of Japan