

Security Automation and Continuous Monitoring
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

J. Li
J. Fu
X. Li
Y. Cheng
China Mobile
November 02, 2020

light weighted vul record
draft-li-sacm-light-weighted-vul-record-00

Abstract

Vulnerability information will be recorded in risk detection and scanning. If a vulnerability is detected in a host during one scan, a record will be generated and added to the database, together with a time-stamp when the vulnerability is detected. If risk detection is carried out periodically, a series of records will be generated for each detection, until vulnerability is fixed. At present, a common way to record vulnerabilities is a vulnerability--a detection--a record, a vulnerability--N detections--N records($N > 1$). In this way, the number of vulnerability records is related to the rounds of detection. In the case that the number of existing vulnerabilities remains unchanged, more frequent vulnerabilities are scanned, more records are recorded. In this document, a light weighted vulnerability recording method is proposed. To make that, in the whole life cycle of a vulnerability, only one record is generated after multiple detections.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119][RFC8174].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 6, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Data Structure	3
3.	Normative References	6
	Authors' Addresses	6

[1.](#) Introduction

In the process of risk management, large enterprises and institutions often use the method of periodic vulnerability detection and scanning. At present, the common way to record vulnerabilities is to keep many records for each vulnerability. Since the vulnerability state is time-dependent, each vulnerability record will carry time information to indicate the time stamp when the vulnerability is detected. A vulnerability record usually contain the following elements:

- 1.IP address
- 2.port: e.g. 23
- 3.service: e.g. telnet
- 4.protocol: TCP or UDP
- 5.vul_name: e.g. telnet weak password

6.vul_det_time: time stamp when the vulnerability is detected, e.g.
2020-10-10 10:20:30

7.vul_detail: additional information of vulnerability, in the
scenario of weak password, vul_detail could be admin / 123456

By using these data, security administrators can locate the
vulnerabilities on specific hosts and specific services.

2. Data Structure

In the work of continuous security monitoring, if the same
vulnerability of the same host is not repaired in time, multiple
records will be generated. And the more frequent periodic detection,
the more vulnerability records will be generated. Repeated
vulnerability records will waste storage space and increase the cost
of data analysis. This draft proposes a light weighted vulnerability
recording method, which stores only one record for the same
vulnerability of the same host. By optimizing the structure of data,
a record contains a relatively rich life process of vulnerabilities.

Light weighted vulnerability record defined in this document contains
the following elements:

IP, port, service, protocol, vul_name, vul_status, vul_det_time,
vul_update_time, vul_fix_time, vul_detail.

Compared with the vulnerability record above, the light weighted
vulnerability record adds 3 elements:

1.vul_status: show the vulnerability is currently existing or fixed.
0 indicates that the vulnerability has been fixed, and 1 indicates
that the vulnerability exists. For those vulnerabilities detected
for the first time, vul_status is 1, and for those previously
detected but currently fixed vulnerabilities, vul_status is 0

2.vul_det_time: time stamp that detect the vulnerability first time.
3. Vulnerability repair time: when a vulnerability is found to have
been fixed during a certain detection, the time at that time will be
recorded

3.vul_fix_time:once it is detected that the vulnerability has been
fixed, record this time.

Compared with the vulnerability record above, the light weighted
vulnerability record re-defined 1 element:

1.vul_update_time:last update time of vulnerability status.
corresponding to the previous vul_update_time, it has different meaning from the previous one. It represents the latest update time of vulnerability status. When detect many times, the latest time stamp will be recorded to cover the old record.

Suppose that a company carried out a round of vulnerability detection every day, and found that there is a telnet weak password on a host from 2020-10-11 to 2020-10-17. And it was fixed on 2020-10-18. One vulnerability record generated everyday. Records were as follows:

```
{  
  
IP:A.B.C.D,  
  
port:23,  
  
service:telnet,  
  
protocol:TCP,  
  
vul_name:telnet weak password,  
  
vul_detect_time:2020-10-11 10:20:30,  
  
vul_detail:admin/123456  
  
}
```

{Records for 2020-10-12}

{Records for 2020-10-13}

{Records for 2020-10-14}

{Records for 2020-10-15}

{Records for 2020-10-16}

```
{  
  
IP:A.B.C.D,  
  
port:23,  
  
service:telnet,  
  
protocol:TCP,
```



```
vul_name:telnet weak password,  
vul_detect_time:2020-10-17 10:20:30,  
vul_detail:admin/123456  
}
```

On 2020-10-18, no more records for this vulnerability, since it was fixed.

By using light weighted vul record mentioned in this document, this telnet weak password generated only one record. On 2020-10-17, record was as follow:

```
{  
IP:A.B.C.D,  
port:23,  
service:telnet,  
protocol:TCP,  
vul_name:telnet weak password,  
vul_status:0,  
vul_det_time:2020-10-11 10:20:30,  
vul_update_time:2020-10-17 10:20:30,  
vul_fix_time:2020-10-17 10:20:30,  
vul_detail:admin/123456  
}
```

On 2020-10-18, record was as follow:

```
{  
IP:A.B.C.D,  
port:23,  
service:telnet,
```



```
protocol:TCP,  
vul_name:telnet weak password,  
vul_status:0,  
vul_det_time:2020-10-11 10:20:30,  
vul_update_time:2020-10-18 10:20:30,  
vul_fix_time:2020-10-18 10:20:30,  
vul_detail:null  
}
```

3. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Jiang Li
China Mobile
Beijing 100053
China

Email: lijiang@chinamobile.com

Jun Fu
China Mobile
Beijing 100053
China

Email: fujun@chinamobile.com

Xiaoxiao Li
China Mobile
Beijing 100053
China

Email: lixiaoxiao@chinamobile.com

Yexia Cheng
China Mobile
Beijing 100053
China

Email: chengyexia@chinamobile.com

