Intrusion Alert Normalization method using AWK scripts and attack name database

Dongyoung Kim, HyoChan Bang, Jung-Chan Na
Electronics and Telecommunications Research Institute
Kdy63281@etri.re.kr, bangs@etri.re.kr, njc@etri.re.kr

The current several classes of intrusion alert have various formats and semantics. And it is transferred using a variety of protocols. The protocols that transfer intrusion alert are IDXP, SNMP trap, SYSLOG protocol, etc. These varieties of intrusion alert formats make it difficult to use that together. Intrusion alert normalization makes various intrusion alert to same structure data and same semantics. We need this normalization process to unify alerts from a variety of security equipments. This paper describes how to normalize alerts from several IDS and security equipments.

Keywords — **Normalization, IDMEF, IDXP, CVE, AWK**

1. Introduction

In network, especially internet, perfect security doesn't exist because of its structure. The best makes low intrusion possibility, detect intrusion quickly, reduce damage, and protect data using backup. Because finding intrusion quickly is more important than any other procedure, Intrusion Detection System (IDS) is the largest part of network security tools.

IDS is a set of tools in relation to detecting unauthorized network access and system use. Intrusion is done by attacker through internet, and it includes obtaining unauthorized system privilege by system user and it includes that system user misuse granted system privilege. IDS is a general term about what is made from software or hardware and what have automated intrusion detection process.

Several IDS have been used, each IDS have been developed their own intrusion detection alert formats and its transfer protocol. Because of the variety of intrusion alert formats and transfer protocols, the interaction of several IDS is complicated.

So standard data exchange protocol and data formats is need, for information sharing IDS.

2. IETF & IDWG

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. The actual technical work of the IETF is done in its working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). Much of the work is handled via mailing lists. The IETF holds meetings three times per year.

The Intrusion Detection Working Group (IDWG) is one of the working groups in the security area. The purpose of the IDWG Group is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems, and to management systems which may need to interact with them. The Intrusion Detection Working Group will coordinate its efforts with other IETF Working Groups.

IDWG has produced a requirements document, alert data formats document (IDMEF), two transport protocols (IAP and IDXP), and tunnel protocol (TUNNEL).

3. IDMEF, IDXP, & CVE

3.1 IDMEF

The Intrusion Detection Message Exchange Format (IDMEF) is intended to be a standard data format that automated intrusion detection systems can use to report alerts about events that they deem suspicious. The development of this standard format will enable interoperability among commercial, open source, and research systems, allowing users to mix-and-match the deployment of these systems according to their strong and weak points to obtain an optimal implementation.

The most obvious place to implement the IDMEF is in the data channel between an intrusion detection analyzer (or "sensor") and the manager (or "console") to which it sends alarms.

IDMEF messages are XML (Extensible Markup Language) documents that notice attack trial. In IDMEF document, there are intrusion alert expressions, IDMEF data types, and so on.

IDMEF messages have one or more alert, and <IDMEF-Message> is a it's top tag. This tag has <Alert> or <Heartbeat> tag. <Alert> has analyzer name that exports alert, event, source, target information. And <Heartbit> pass on state information from analyzer to manager.

3.2 IDXP

IDXP is application level protocol that transfer data among intrusion attack detection entity, and specified, in part, as a Blocks Extensible Exchange Protocol (BEEP) "profile".

BEEP is a generic application protocol framework for connection-oriented, asynchronous interactions. It makes new connection-oriented application layer protocol developed fast. Features such as authentication and confidentiality are provided through the use of other BEEP profiles. Accordingly,

many aspects of IDXP (e.g., confidentiality) are provided within the BEEP framework.

IDXP provides for the exchange of IDMEF messages, unstructured text, and binary data between intrusion detection entities.

Addressing the security-sensitive nature of exchanges between intrusion detection entities, underlying BEEP security profiles should be used to offer IDXP the required set of security properties.

IDXP is primarily intended for the exchange of data created by intrusion detection entities. IDMEF messages should be used for the structured representation of this intrusion detection data, although IDXP may be used to exchange unstructured text and binary data.

Before IDXP, the working group designed Intrusion Alert Protocol (IAP) as protocol for intrusion alert exchange. The design of IAP was based on the Hypertext Transfer Protocol (HTTP). However the working group decided it should investigate the Blocks Extensible Exchange Protocol as the basis for the IDWG transport protocol. IAP was alternated by IDXP and expired.

3.3 CVE

Common Vulnerabilities and Exposures (CVE) is a list or dictionary that provides common names for publicly known information security vulnerabilities and exposures. Using a common name makes it easier to share data across separate databases and tools that until now were not easily integrated. This makes CVE the key to information sharing. If a report from one of your security tools incorporates CVE names, you may then quickly and accurately access fix information in one or more separate CVE-compatible databases to remediate the problem.

Most information security tools include a database of security vulnerabilities and exposures; however, there is significant variation among them and no easy way to determine when different databases are referring to the same problem. The consequences are potential gaps in security coverage and no effective interoperability among the disparate databases and tools. In addition, each tool vendor currently uses different metrics to state the number of vulnerabilities or exposures they detect, which means there is no standardized basis for evaluation among the tools.

With a standard list of vulnerabilities and exposures such as CVE, your databases and tools can "speak" to each other. And, you'll know exactly what each tool covers because CVE provides you with a baseline for evaluating the coverage of your tools. This means you can determine which tools are most effective and appropriate for your organization's needs. In short, CVE-compatible tools and databases will give you better coverage, easier interoperability, and enhanced security.

CVE is also endorsed by leading representatives from the information security community. CVE's content results from the collaborative efforts of the CVE Editorial Board, which includes representatives from numerous information security-related organizations.

The MITRE Corporation maintains CVE and provides neutral guidance to the Editorial Board on all matters related to the ongoing development of CVE. In partnership with government, MITRE is an independent, not-for-profit corporation working in the public interest. It addresses issues of critical national importance, combining systems engineering and information technology to develop innovative solutions that make a difference.

3.4 The advantage of standard

These standards offers three advantages like the next statement.

- The ease of working with each other
  Because of using same alert format and transfer protocol, the security equipments from various vendors speak each other.
- Correlation
  Because Routers, IDS, Firewalls, and other security equipments use IDMEF, it is easy to correlate events from various sources.
- The easy arrangement of security equipments
  It is easy to arrange different vendor's IDS together, and to change the part of IDS.

But current most IDS use their own alert formats and protocol for alert exchange. For using these various IDS together, it is necessary to support various protocols. And it is necessary to normalize various formats of alert to unified information two.

4. System Configuration

Our Normalization System is a component module of intrusion alert collector system. Intrusion alert collector gather intrusion alert from several security equipments including IDS and normalize it. Normalized results are passed to network attack analyzer and saved to database. Figure 1 shows the scenario using intrusion alert collector.

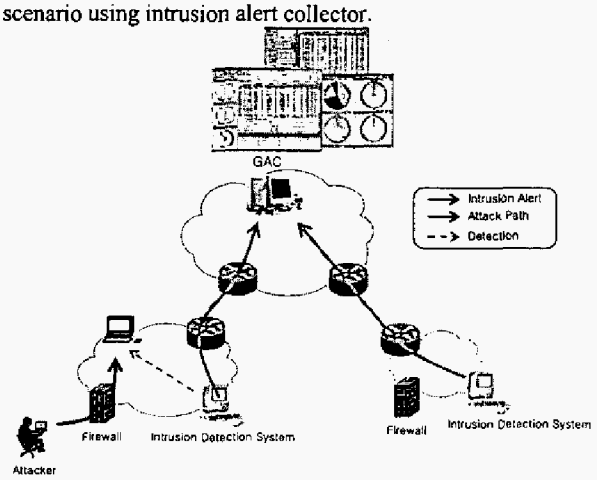

Figure 1. scenario using intrusion alert collector

System collects intrusion alerts through IAP, IDXP, BSD SYSLOG protocol, and SNMP trap. The collected alert Formats are IDMEF, SYSLOG message, trap. These alerts are normalized, changed to same type information, and written to queue. That information are read from queue, passed to

network attack analyzer and written to database. Figure 2 shows intrusion alert collector's system structure.

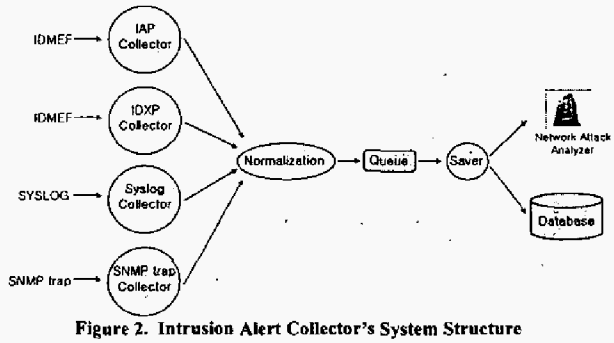

Figure 2. Intrusion Alert Collector's System Structure

5. The Four Stages of Normalization

Our alert normalization method is divided into two parts. First part is Syntax Normalization that get unified alert information from various formats of alerts. And second part is Semantic Normalization that get unified attack name (or vulnerability name) and attack severity. First part is divided into three stages. So our alert normalization method have four stages.

Figure 3 shows four stages of normalization.

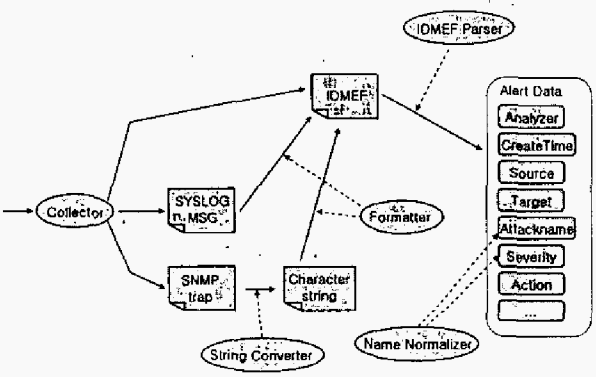

Figure 3. Four Stages of Normalization

5.1 The first stage: converting to string

First stage is to change all alert into character string. The alert through SNMP trap is changed to character string by writing OID and value of MIB. If alert format is IDMEF or simple character string, first stage is skipped.

If the alert is received through SNMP trap, we make it pairs of OID and value. The converted traps looks like next shape.

"oid=value, oid=value..."

The next string is an example.

0,public,.1.3.6.1.4.1.7424.1=10.10.30.23,255,0,.1.3.6.1.4.1.7424.1.3.1=[index 189625] 04/09/21 15:30:41 (src 10.10.70.11) (dst 10.10.30.23) (icmp: 8, 0) [icmp ping bsd type] <sensor 10.10.30.23>,

5.2 The second stage : making IDMEF

The second stage is to change character string into IDMEF message using AWK scripts. IDMEF is defined by IDWG working group of IETF. The purpose of IDMEF is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems, and to the management systems which may need to interact with them.

The AWK is a text-processing language. It has two faces: it is a utility for performing simple text-processing tasks, and it is a programming language for performing complex text-processing tasks. Several kinds of tasks occur repeatedly when working with text files. You might want to extract certain lines and discard the rest. Or you may need to make changes wherever certain patterns appear, but leave the rest of the file alone. Writing single-use programs for these tasks in languages such as C, C++, or Pascal is time-consuming and inconvenient. Such jobs are often easier with AWK. The AWK utility interprets a special-purpose programming language that makes it easy to handle simple data-reformatting jobs.

Figure 4 shows to convert from SNORT alert message into IDMEF message.

```
<4> snort: [1:1418:2] SNMP request tcp [Classification: Attempted Information Lea
k] [Priority: 2]: {TCP} 192.168.0.73:26554 -> 10.10.30.4:80
```

```
<IDMEF-Message version="1.0"> <Alert>
<Analyzer analyzerid="" model="SNORT">
<Node><Address category="ipv4-addr"><address></address></Address></Node>
</Analyzer>
<Source>
<Node><Address category="ipv4-addr"><address>192.168.0.73</address>
</Address></Node>
<Service><port>26554</port></Service>
</Source>
<Target>
<Node><Address  category="ipv4-addr"><address>10.10.30.4</address></Address>
</Node>
<Service><port>80</port><protocol>TCP</protocol></Service>
</Target>
<Classification origin="vendor-specific"><name>1418</name>
<url>http://www.snort.org/</url></Classification>
</Alert> </IDMEF-Message>
```

Figure 4. converting from SNORT alert into IDMEF

Figure 5 shows AWK scripts for converting into IDMEF message. There are several other scripts according to alert formats. If we support new alert format, we will add new AWK script to change it into IDMEF message.

```
BEGIN {
    parse = 0
}

# SNORT
/snort:/ && /\[Classification:/ && /\[Priority:/ {
    parse = 1
    .....
    for ( i = 1 ; $i !~ /\]/ ; i++ )
    count = split($i, temp1, ":")
.....

}

# TESS
/.1.3.6.1.4.1.7424.1/ && /.1=\w+[index/ {
    parse = 1
    analyzer_id = ""
    analyzer_model = "TESS"
    .....
}

END {
    if ( parse > 0 ) {
        message = sprintf("<IDMEF-Message version=\"1.0\"><Alert>
        .....
        </Alert></IDMEF-Message>",
        analyzer_id, analyzer_model, analyzer_addr,
        .....
    }
}
```

Figure 5. AWK scripts to convert alert to IDMEF message

There are several other scripts according to alert forms. If we support new alert format, we will add new AWK script to change it into IDMEF message.

5.3 The third stage : parsing IDMEF message

Third stage is to get alert data from IDMEF message using IDMEF XML parser.

Figure 6 shows a sample alert data from IDMEF message.

```
<IDMEF-Message version="1.0"> <Alert>
<Analyzer analyzerid="" model="SNORT">
<Node><Address category="ipv4-addr"><address></address></Address></Node>
</Analyzer>
<Source>
<Node><Address category="ipv4-addr"><address>192.168.0.73</address>
</Address></Node>
<Service><port>26554</port></Service>
</Source>
<Target>
<Node><Address category="ipv4-addr"><address>10.10.30.4</address></Address>
</Node>
<Service><port>80</port><protocol>TCP</protocol></Service>
</Target>
<Classification origin="vendor-specific"><name>1418</name>
<url>http://www.snort.org/</url></Classification>
</Alert> </IDMEF-Message>
```

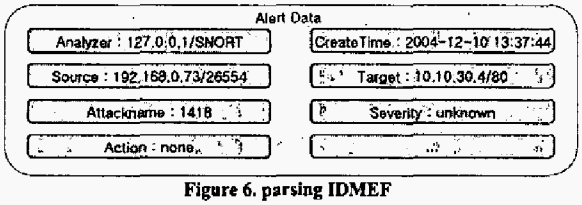

Figure 6. parsing IDMEF

5.4 The fourth stage : Semantic Normalization

The fourth stage is to get normalized attack name and severity. Several IDS use different attack name about same attack situation. But it is necessary to make same name for we analysis these alerts together. Name Normalization change IDS's attack name into CVE name. CVE is a list of standardized names for vulnerabilities and other information security exposures, CVE aims to standardize the names for all publicly known vulnerabilities and security exposures. If the informations about CVE name related IDS's attack name exist, We make it database. The semantic normalization stage is to search related CVE Name from Name Normalization database. Figure 7 shows semantic normalization stage.

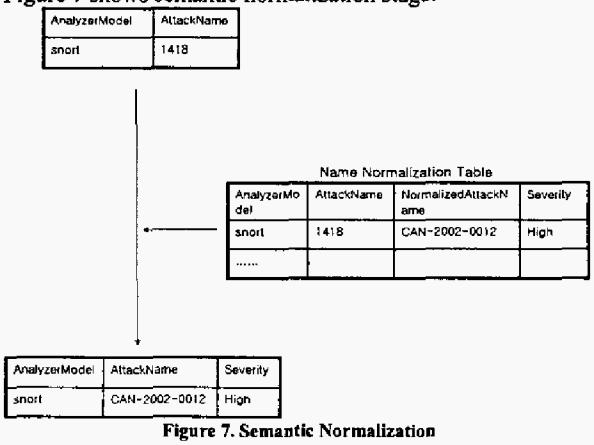

Figure 7. Semantic Normalization

6. Summary & Future Work

This paper describes IDWG that standardize alert format and exchange protocol and the standards related with intrusion alert. And it explains the way how we normalize alerts from several IDS.

We plan to add Firewall log normalization, and make syntax normalization method without AWK script about alert formats used frequently.

REFERENCES

[1] J.Betser, A. Walther, M.Erlinger, T.Buchheim, B, Feinstein, G. Matthews, R. Pollock, K. Levitt , , and K. Levitt, "GlobalGuard: Creating the IETF-IDWG Intrusion Alert Protocol(IAP)", 2001.

[2] Tim Buchheim, Michael Erlinger, Ben Feinstein, Greg Matthews, Roy Pollock, Joseph Betser, and Andy Walther, "Implementing the Intrusion Detection Exchange Protocol", Dec, 2001.

[3] M. Wood, and M. Erlinger, "Intrusion Detection Message Exchange Requirements", Oct, 2002. draft-ietf-idwg-requirements-10

[4] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format", July, 2004. draft-ietf-idwg-idmef-xml-12

[5] B. Feinstein, G. Matthews, and J. White, "The Intrusion Detecting Exchange Protocol (IDXP)", Oct. 2002. draft-ietf-idwg-beep-idxp-07

[6] http://www.ietf.org/html.charters/idwg-charter.html

[7] http://www.cve.mitre.org/