# Effective Security Monitoring through System Recognition

Felix von Eye
Leibniz Supercomputing Centre, Munich Network Management Team
Garching n. Munich, Germany
Email:voneye@lrz.de

***Abstract*—The bottleneck of security monitoring is often the huge amount of signatures, which are useless but consume computation power and time. Therefore, the signatures have to be set more accurate for the systems, which should be protected. In this paper, a new approach is presented, which is able to detect more efficient the service and software running on a server. This knowledge helps to select the relevant signatures for the security monitoring, which leads to a more efficient usage of the system's resources.**

***Keywords–security monitoring; network security; security management; proactive scan; netflows; flow records.***

## I. INTRODUCTION

Security threats are a challenge for every IT infrastructure. To deal with this threat, there are a lot of different approaches introduced in the past. In general there are three different categories of detection: signature based, anomaly or behavior based, and visualization based [1]. In real world scenarios, only signature based detection systems are widely used, because of the low rate of false positive alarm messages. The anomaly based detection is often used in research or at anti malware companies as this method leads to the detection of new and unknown attacks and malware software. Last but not least, the visualization of system behavior to recognize are at most used in network operation centers with the focus on detection of network anomalies.

In the following, the paper focuses on signature based detection systems. The main drawback of these solutions is that they have to carry all the signatures, such as the ones from the actual threats and all the past signatures. The reason for this is that is is possible that an attacker to use an old vulnerability to penetrate a system, e.g., the system administrator installs an older vulnerable software or the outdated exploit is functional in other constellations as well.

In intrusion detection systems, i.e., Snort [2], there are tens of thousands of rules for different attack and intrusion scenarios [3]. A similar situation is visible in the analysis of virus scanners. There are at the moment more than ten million signatures, which can detect viruses, trojan horses or other malware [4]. These signatures are defined by analyzing software and attack behavior, e.g., the communication of a bot software with its command and control server.

On one side, it is pleasing that the security scanners support a widespread malware detection but on the other hand, the scanners have to scan through this huge amount of signatures to evaluate a threat. By implication, the more signatures are added to the scanners databases the slower it works. This leads to the paradox situation that security officers have to pick only the signatures they hope to be the most critical ones. But how should they know? As the other signatures are not activated they cannot find anything related to the deactivated signatures.

To deal with this problem, this paper proposes a new concept of choosing signatures in security monitoring. This paper focuses without loss of generality at most parts network security monitoring. In other parts of the security monitoring, the knowledge acquisition, e.g., the system recognition, differs, whereas the underlying ideas remain the same. First of all, it is important to know which systems are running and what services they offer. This information is very important for the second step. In this step, the systems are categorized, which enables the possibility to choose only these signatures, which are related to this category, e.g., if there is an apache web server running, only the signatures related to apaches or/and web servers are activated.

The rest of the paper is structured as follows. In the next Section II we take a look at different methods to gain information about a system. These methods are combined in Section III. The results of this service detection is discussed in Section IV. At the end, Section V gives a short conclusion and an outlook on next steps.

## II. METHODS OF SYSTEM AND SERVICE DETECTION

In general, there are four different methods to detect systems and services [5]. On one side, there are passive methods and on the other hand, there are active methods. In both categories, there are intrusive methods and also non intrusive ones. In the following, these methods are presented.

### A. Active and intrusive asset detection

The active methods are used by administrators, who directly want to know something about a system and therefore scans the system. By using intrusive methods, the administrators take advantage of the fact that every system contains bugs and security holes. These bugs are mostly unique, so it is possible to determine, what is currently running. For example, if you are able to exploit the vulnerability *CVE-2015-0929* [6], than it is clear that the system is a SerVision HVG Video Gateway with firmware before 2.2.26a78. With more knowledge it is possible to examine more precisely the used firmware or software version.

A collection of usable exploits is for example the metasploit framework [7], which includes in total several thousands of different exploits.

The main drawback is that the usage of vulnerabilities is a very high risk for the system, which should be scanned. Very often, vulnerabilities leads to service faults or crashes. On the

 ISBN: 978-1-61208-413-8 

other hand, for every administrator in charge of security it should be best practice to patch a vulnerable system as soon as possible, so it is not impossible that after a short period of time, this scanning method doesn't get any results.

### B. Active and non intrusive asset detection

In the other active method, the non intrusive active method, the administrator scans a system by using port scanners like nmap [8]. These scanners send some predefined network packets to the other system and wait for the response. Because there are some differences in the implementation between different operating systems, it is possible to determine the fingerprint of each system by analyzing the response packets. With this method, regular scans can be performed also to detect changes in the configurations, e.g., via the tool Dr. Portscan [9].

With this method, it is possible to determine, what services are running on a specific system, but as there are in general no differences between the versions in regard of the way of response, it isn't possible to determine the correct version of a service or operating system. Often there are hello messages from the services, in which they identify themselves, but this information can be set by the local administrator without any side effect and is therefore not trustworthy.

But as pointed out by Mäurer in [10], also port scans can be hazardous for servers, particularly if the port scan is done very fast, e.g., with the tool masscan [11] or Zmap [12], which is able to scan a system with more than 10 Gbps.

### C. Passive and non intrusive asset detection

In opposite to active methods, the passive methods use only data, which can be measured in the communication path, e.g., at a router or switch. The common way is to use flow records based on the netflow protocol to determine, which host has a connection to other systems [13]. In general, flow records don't contain any information about the services, but with some assumptions it is possible to determine, what kind of service is hosted on a system. For that, it is helpful that the Internet Assigned Numbers Authority (IANA) reserved a huge amount of ports for specific programs, e.g., in Transmission Control Protocol (TCP) port 22 for the Secure Shell (SSH) service. If there is a connection between two hosts with port 22 involved, the probability is very high that there is really a SSH server running on one host.

This method has no effect on the connection or the availability of the service. On the other hand, it is only possible to detect services and systems, which are really communicating.

A very interesting challenge is to determine which operating system is running on one specific system. Therefore, it is needed to know, which update server are normally used in a environment. In general, the update servers of the Microsoft Windows operating system are in the IP range 65.55.0.0/16, so if one finds a connection between a system and these IP addresses and there is more traffic visible than only a port scanning, then it is likely a Windows system. A little bit more complicated is, if you have on Windows side some Windows Server Update Services (WSUS) running or you have Linux systems, which have possibly different update repository servers installed, or you have systems, which are never or only offline updated.

As there are many services, which have their one update servers, it is possible to have a huge list of potential update

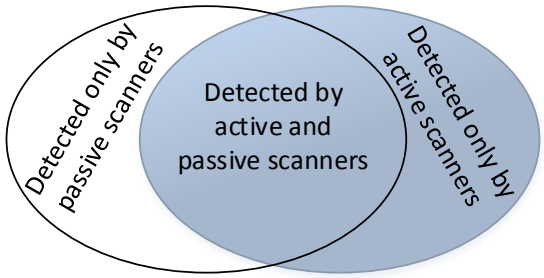


Figure 1. Venn diagram of service detection.

communication, which helps to verify the analysis of the other flow record communication. But it is not possible to determine the version of the used software, as this information isn't transmitted anywhere.

### D. Passive and intrusive asset detection

The other possibility to use passive methods is the passive intrusive asset detection, which is done in general via deep packet inspection. In the deep packet inspection, the content of the communication packets are inspected, which enables a very detailed analysis of the communication partners, under the constraint that the communication isn't encrypted with a strong cipher [14]. Similarly, to the port scanners, also at the deep packet inspection it is possible to use some fingerprints to connect the observed communication with a specific service.

This method has no effect on the connection or the availability of the service. On the other hand, it is only possible to detect services and systems, which are really communicating.

As already mentioned, it is difficult to handle encrypted data, but in some cases there are some hints inside the packets, which enable the classification of the traffic without the knowledge about the content of the messages. Furthermore, there are especially in the German and European countries legal constraints in regard of privacy, which don't allow providers to analyze the content of a communication [15].

## III. Vulnerability management for rules creation

As mentioned before, the most exact method to determine the right type and version of a software is the vulnerability scan, which leads unfortunately sometimes to a system instability. On the other hand, flow record analysis in combination with deep packet inspection enables to detect the fewest amount of software type and in general no version information. But because of routing or firewall restrictions it is most likely that the active scanners are not able to reach any system or service in a network, as passive scanners can only see services, which are used in the detection time frame. Figure 1 shows the overall set of problems in the detection rate.

The biggest challenge in passive detection is that there is no correlation, which enables the administrators to connect the observation of a used connection with the corresponding service. Therefore, we propose the usage of vulnerability scanners to improve the detection on flow record basis. This is achieved with the following steps.

In a first step, all systems are scanned with a vulnerability scanner with a small and controlled rule set. This leads to a

 ISBN: 978-1-61208-413-8 

list of services and systems, which is reliable. At this point, it is very important that the exploits, which are used in this step, are very well tested to minimize the risk for the connected systems. Furthermore, it is helpful, if the used exploit is not very old. Otherwise, most systems and services are patched by the local administrators in the meantime. But as studies have shown, even for well reported vulnerabilities as the POODLE vulnerability [16], there are four months later even 25 percent of the tested systems still not patched in a bigger university research network [10].

This list of systems and services is now taken to feed the flow record analysis step. From this time on, the flow records of these systems are recorded. As even one service on the host is well known, it is interesting to determine, if the communication from this service can be made visible in the flow records. On the other hand, it is possible to use the knowledge of the deep packet inspection, which allows to determine the used protocol, e.g., with the tool nDPI [17]. This tool is able to detect, for example if a Hypertext Transfer Protocol (HTTP) protocol is used in the communication, even if not the standard port 80 is used. As the deep packet inspection is not able to determine the service, there could be a Linux based Apache web server or a Windows based IIS or other combinations by monitoring a HTTP connection. Nevertheless, the deep packet inspection is very helpful to filter the relevant flow records.

As the relevant flow records are now found, the next step is to compare them in a way, to find some characteristics. So it is noticeable that the response time of an IIS web server is smaller as the same request on an Apache web server. So it is possible to distinguish these two software products by only looking at flow records. As the response time of a web server is in general dependent from the amount of connections per minute, there has to be at least a relationship between the amount of connections and the response times. A fixed value would not be adequate in this case.

Other criteria are the amount of responses to one request, the size of the packets, the delay between packets or also the first bytes of the content of the packet. Depending on the used software, it is possible to detect it only with the usage of deep packet inspection for the used protocol and also flow records for the characteristics.

## IV. Using detection results

The results in the detection stage can now be used for the security monitoring. On one side, it is now possible to improve the security incident management [18], by providing more information about a system and the possible break in. On the other hand it is also possible to reduce the total amount of signatures, which are to be used in the security monitoring.

If only the necessary signatures are used, it is possible for a security administrator to use in total more signatures on the same hardware, which leads to total increase of overall security.

In most environments, security monitoring, e.g., network intrusion detection, is done by splitting the traffic on several monitoring points. Mostly the separation is done on port based rules, so all traffic on port 80 is analyzed at one security monitoring point, while the traffic on port 22 is analyzed on another monitoring point. This method is not working very well for huge heterogeneous networks, as not every web server is running on port 80 or 443. The proposed service detection leads to a more exact assignment of systems to monitoring points.

To reach this, all traffic is classified with the above presented method. As the classification allows to determine the exact service, the security monitoring can be adjusted accurately fitting. On the other hand, the monitoring of the flow records and also the deep packet inspection for the protocol detection is not very expensive for monitoring systems.

A very important aspect is, that in regular intervals new vulnerabilities and therefore new classifications are put into account. Due to software updates or changes in protocols and implementations, it is possible that the characteristics are changing.

The main drawback of this approach is the delay until new rules and assignments can be activated. Furthermore, flow records are generally exported regularly after a short amount of time from the switch and analyzed offline, so there is also a measurable delay. A solution could be to transform the approach to a northbound controller application in Software Defined Networks (SDN). There is the possibility that the export of flow records is not needed anymore, as the detection is done by a northbound application at the network controller. This can reduce significantly the delay of the analysis step. But besides from the improvements of using SDN, it is in general unusual that systems and services change very often. Often, there are only small deltas in regard of the different usage of systems, which indicates that the delay is not very important in real world scenarios.

## V. Conclusion and Outlook

In this paper, a new approach for the assignment of signature based security monitoring is proposed. This propose was based on the knowledge of vulnerabilities to create rules, which are applicable for flow record detection. This allows administrators to improve the security monitoring of their domain.

In the next steps,we need to define how vulnerabilities, which are used in this approach for crosschecking the results in the rule creation phase, can be processed in an automatic manner. Furthermore, there have to be discovered more characteristics of different services, which allow to use this approach in a wider domain.

## Acknowledgments

Parts of this work has been funded by the German Ministry of Education and Research (FKZ: 16BP12309). The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful comments on previous versions of this paper. The MNM-Team, directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering, is a group of researchers at Ludwig Maximilian University of Munich, Technische Universität München, the University of the Federal Armed Forces, and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities.

## References

[1] Y. Jiang, Z. Jin, A. Abdelkefi, M. Ask, and H. Skrautvol, “Network anomaly detection through traffic measurement,” in Annual Report 2010, S. J. Knapskog, Ed. Trondheim, Norway: Centre for Quantifiable Quality of Service in Communication Systems, Apr. 2011.

[2] M. Roesch. Snort – The Open Source Network Intrusion Detection System. [Online]. Available: http://www.snort.org [retrieved: May, 2015]

[3] G. Münz, N. Weber, and G. Carle, "Signature detection in sampled packets," in Workshop on Monitoring, Attack Detection and Mitigation (MonAM 2007), Toulouse, France, 2007.

[4] F-Secure Corporation. F-Secure DeepGuard™2.0. Helsinki, Finland. [Online]. Available: https://www.f-secure.com/system/fsgalleries/white-papers/f-secure_deepguard_2.0_whitepaper.pdf [retrieved: May, 2015]

[5] A. Bernhard, "Netzbasierte Erkennung von Systemen und Diensten zur Verbesserung der IT–Sicherheit," Bachelor Thesis, Ludwig Maximilian University of Munich, Munich, Mar. 2014.

[6] NIST – National Vulnerability Database. SerVision HVG Video Gateway web interface contains multiple vulnerabilities. [Online]. Available: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-0929 [retrieved: Mar., 2015]

[7] Rapid 7. Penetration Testing Software – Metasploit. [Online]. Available: http://www.metasploit.com [retrieved: May, 2015]

[8] G. Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Sunnyvale, USA: Insecure.Com, 2008.

[9] W. Hommel, S. Metzger, D. Pöhn, and F. von Eye, "Improving higher education network security by automating scan result evaluation with Dr. Portscan," in ICT Role for Next Generation Universities, ser. EUNIS 2014 – 20th EUNIS Congress, U. Sukovskis, Ed., Umea, Schweden, Jun. 2014, pp. 73–83.

[10] N. Mäurer, "Efficient scans in a research network," Bachelor Thesis, Technische Universität München, Munich, Feb. 2015.

[11] R. D. Graham and P. C. Johnson, "Finite state machine parsing for internet protocols: Faster than you think," in Security and Privacy Workshops (SPW), 2014 IEEE, May 2014, pp. 185–190.

[12] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-wide scanning at 10 Gbps," in Proceedings of the 8th USENIX conference on Offensive Technologies. USENIX Association, 2014, pp. 1–8.

[13] M. E. Klepsland, "Passive Asset Detection using NetFlow," Master Thesis, University of Oslo, Department of Informatics, Oslo, Feb. 2012.

[14] T. Böttger, "Detection of Amplification Attacks in Amplifier Networks," Master Thesis, Technische Universität München, Munich, May 2014.

[15] F. von Eye, W. Hommel, and D. Schmitz, "A Secure Logging Framework with Focus on Compliance," in International Journal on Advances in Security, R. Savola, Ed., vol. 7, no. 3 & 4. IARIA, Dec. 2014, pp. 37–49. [Online]. Available: http://www.iariajournals.org/security/sec_v7_n34_2014_paged.pdf

[16] B. Möller, T. Duong, and K. Kotowicz, "This POODLE Bites: Exploiting The SSL 3.0 Fallback." Google, Sep. 2014.

[17] ntop. nDPI – Open and Extensible LGPLv3 Deep Packet Inspection Library. [Online]. Available: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-0929 [retrieved: May, 2015]

[18] S. Metzger, W. Hommel, and H. Reiser, "Integrated Security Incident Management – Concepts and Real-World Experiences," in 6th International Conference on IT-Security Incident Management and IT Forensics, Stuttgart, May 2011.

 ISBN: 978-1-61208-413-8 