Improving higher education network security by automating scan result evaluation with Dr. Portscan

Felix von Eye[1], Wolfgang Hommel[1], Stefan Metzger[1], Daniela Pöhn[1]

[1]Leibniz Supercomputing Centre, 85748 Garching, Germany,
{voneye,hommel,metzger,poehn}@lrz.de

Keywords
Network security, portscans, penetration tests, proactive information security management

1. ABSTRACT

In most higher education institutions (HEIs), IT systems are still operated in a decentralized manner at least partially: Although a central data center or IT department typically provides basic IT services such as email servers, many faculties and chairs operate, for example, web servers and file servers on their own. As there often is no campus-wide asset management or configuration management database available, this results in a lack of a big picture, i.e., nobody really knows who is operating which IT services for whom in total throughout the HEI. Problems arise when network services, i.e., servers that can be accessed, e.g., via the Internet, are compromised, either by external attackers or by insider threats. While many faculties succeed in basically setting up their own network service machines, e.g., a web server including a database server for a learning management system, only few of them are aware of typical information security issues and know how to harden their server machine installations to protect them against the usual attacks. To remedy this partial lack of know-how, an increasing number of HEIs set up a central security team tasked with monitoring, analyzing, and continuously improving information security across the campus.

In this article, we present an open source tool, which we developed to facilitate asset management and risk analysis of network services in research and education networks: Dr. Portscan is a delta-reporting tool for network port scans, which are often used for active-probing-based asset discovery, allow for a basic risk assessment, and can be used as a basis for fully-fledged vulnerability management. Dr. Portscan orchestrates the execution of an arbitrary number of port scans, e.g., based on the well-known nmap tool, from various locations inside and outside an HEI's campus network, aggregates and consolidates these port scan results, analyzes which changes have been made compared to the previous state, and can alert the security staff about new or unknown network services on the campus that need more detailed manual investigation.

Dr. Portscan cannot only be used by each HEI individually, but based on agreements between HEIs, security teams at multiple HEIs can collaborate to provide each other with external perspectives on their network services. Dr. Portscan is meanwhile also used in the pan-European CELTIC project SASER – Safe and Secure European Routing – to analyze the basic security properties of active network components, especially routers and network gateways, of the participating internet service providers.

The remainder of this article is structured as follows: In the next section we describe the motivation for the development and use of Dr. Portscan in detail. Section 3 outlines preliminary considerations that should be kept in mind before setting up a technical infrastructure for Dr. Portscan to ensure high quality results. Section 4 presents the technical architecture of Dr. Portscan in detail to provide an understanding of its inner workings, strengths, and limits. Afterwards, Section 5 discusses examples of how Dr. Portscan can be used in practice. Finally, we give an outlook to how Dr. Portscan will be further improved as a part of our ongoing work. Dr. Portscan can be downloaded from: https://git.lrz.de/?p=DrPortScan.git;a=tree

2. MOTIVATION

In many higher education institutions (HEIs) employees are often allowed to install new software packages or change the system's configuration parameters themselves to provide updated and new services, e.g. XMPP-based services like Jabber for communication purposes or any collaboration tools to support project-related tasks. As a result, to keep track of this continuously changing IT-environment is a huge challenge for the responsible security team. Sometimes they do not know immediately after putting to operation which systems are currently running and which services they provide. The system administrators do not announce which new vulnerabilities and risks arise and thus the security team does not know which new or adapted countermeasures must be taken to effectively protect the infrastructure against various network-based attacks.

At the same time, HEIs allow their employees to bring their own devices (BYOD). Thus, employees utilize private tablets or smartphone devices, which are not under direct control of the security people for job-related tasks. Unaware of the risks, which arise if they do this on unencrypted wireless LAN hotspots at the airport these devices, maybe infected by malware or compromised by other means, come back and get connected to the protected networks after their trip.

Another often seen trend is BYOS (bring your own server/service), that mean faculties install and maintain their own server machines at a nearby data center (e.g. server housing) based mostly on economic decision, i.e., security related tasks as patching vulnerable installed software or changing the service configuration are maintained by the external faculty administrators. The internal staff or security team often does not have privileged access to these machines.

Nevertheless, all these scenarios unites that a helpful documentation in the case of a security incident is quite often missing, incomplete or outdated.

To get a more detailed and up to date overview of running systems in a large, heterogeneous network port scans are an adequate means. A port scanner such as nmap (Lyon, 2008) gives this overview and information about connected and available systems, provided services and if TCP/IP

```
Starting Nmap 6.40 ( http://nmap.org ) at 2014-02-21 16:15 CET
Nmap scan report for www.lrz.de (129.187.254.92)
Host is up (0.00021s latency).
rDNS record for 129.187.254.92: www.lrz-muenchen.de
Not shown: 996 filtered ports
PORT     STATE SERVICE    VERSION
80/tcp   open  http       Apache httpd 2
443/tcp  open  ssl/http   Apache httpd 2
8080/tcp open  tcpwrapped
8443/tcp open  tcpwrapped
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1
closed port
Device type: load balancer
Running (JUST GUESSING): F5 Networks embedded (90%)
Aggressive OS guesses: F5 BIG-IP 3650 Local Traffic Manager load balancer (90%)
No exact OS matches for host (test conditions non-ideal).
```

Figure 1: Sample output for a nmap port scan

fingerprinting is enabled additonally about installed operating systems and software versions. But this comes up often by a (very long), unhandy list of open or wrapped ports and system's information as depicted in Figure 1. Addititionally some details are provided based on a guessing step only with a likelihood of more or less than 90 percent.

Nevertheless, the main problem with regular port scans on a large network is the administrative overhead: Depending on the number of servers and services, the list of open ports can quickly scale to a point that is impossible to keep an eye of manually. Watching the whole network requires an accurate, but often very time consuming, analysis of the scan results. In many cases, an organization will not have the personal resources to expend the necessary effort. Additionally, many

organizations lack of a precisely defined, centrally maintained target state against which the scan results could be compared to. Especially this is clearly visible if each individual department sets up their own servers and services.

Another question to answer comes up if the security team or responsible administrators are interested in scanning from different scanner locations to analyze their networks (e.g., one location within the organization and one across the Internet). At this moment, two different views have to be compared and interpreted. Inconsistencies arise because of connections blocked by firewalls or open ports are reported twice if scans occur at different or overlapping points of time. Matching the results requires additional manual analysis and effort.

Furthermore, a port scan only presents a snapshot of a network but the more interesting information belongs to changes and trends of port usage. Each change in the port usage could be an indication of an unintentional misconfiguration, infection by malware or any other system compromise. For instance if a new port gets open on a systems, which provides a web service with an application server and database backend so far and a netflow analysis shows established connections to this port from Internet located machines, then the computer security incident response team should be alerted.

Our work is additionally motivated by the large-scale distributed environment of the SASER-SIEGFRIED project (Safe and Secure European Routing) (Hoffmann, 2014), in which more than 50 project partners design and implement network architectures and technologies for secure future networks. The project's aim is to remedy security vulnerabilities of today's IP layer networks in the 2020 timeframe. Thereby, security mechanisms for future networks will be designed based on an analysis of the currently predominant security problems in the IP layer, as well as upcoming issues such as vendor backdoors and traffic anomaly detection. The project focuses on inter-domain routing, and routing decisions that are based on security metrics. One of the top related problems to SASER-SIEGFRIED is to find new vulnerabilities or malware spreading to have an accurate view on the actual security level. For this, Dr. Portscan is used to find misconfigurations or gateway for malware infection, e.g., a network time service installed on a routing network device or an open SNMP port configuration, which allows amplification attacks.

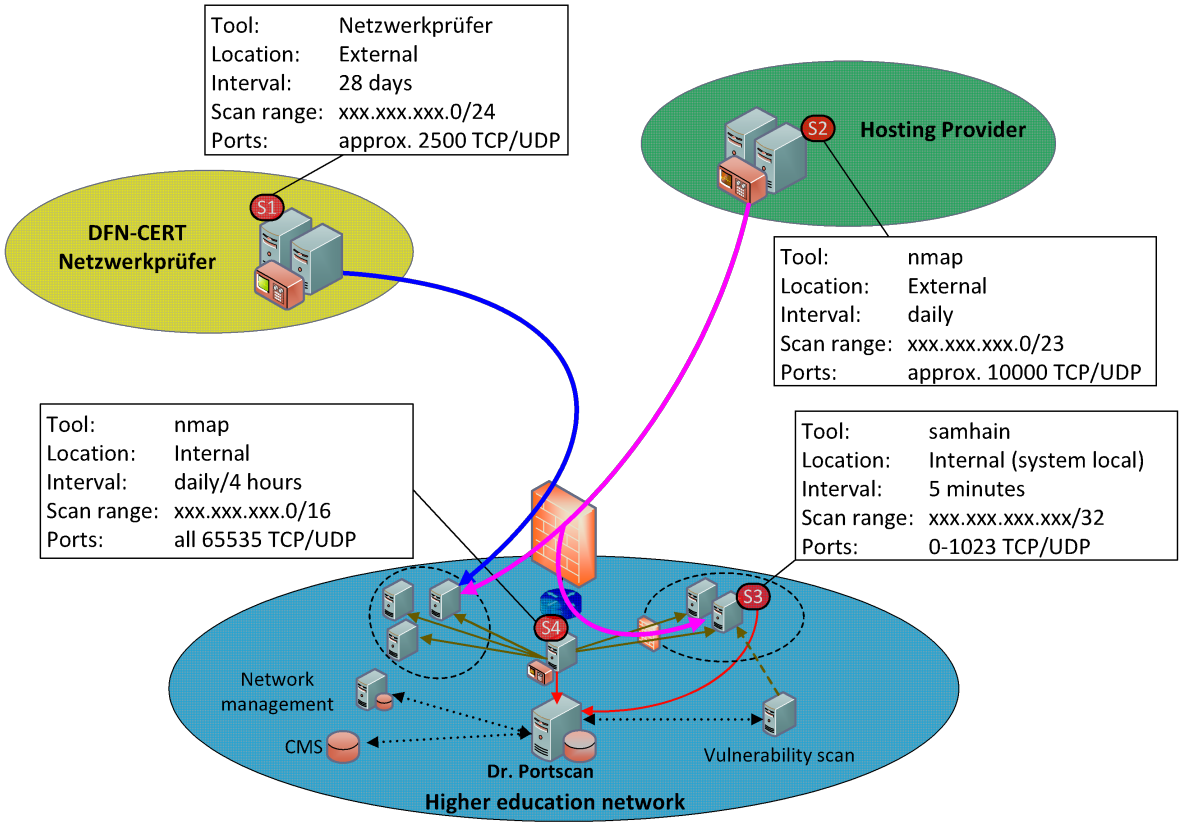


Figure 2: Sample scenario for distributed port scans with Dr. Portscan

3. PRELIMINARY CONSIDERATIONS

Before integrating Dr. Portscan or other tools for scanning the own environment, it is necessary to answer a few questions and to develop a concept, which helps to get authoritative results.

- Which port scanning tools should be deployed? In addition to the classic nmap tool, there are numerous other free and commercial scanners. Security and network management tools can both deliver information about open ports explicitly or implicitly and thus serve as data sources.
- How many systems will run port scans? Even more important, where are the port scanners placed inside the topology? Multiple port scanners at different locations let you differentiate between internal and external views of the networks you analyze but on the other hand, this aspect adds also new complexity steps into the analyzing step. The different lists of results has to be compared and differences has to be highlighted.
- How often are scans from these systems launched? It has to take into consideration that some Internet-based port scanning services only envisage fixed intervals. In addition to the fixed intervals it should be also mentioned that not all systems are running 24/7 so the scanning interval has to be adjusted to the runtime of the machines.
- What is the scope of the scan? External port scans are typically slower than internal LAN scans. For this reason, you might want to restrict scans to individual subnets, or some other subset of all possible TCP/ UDP ports.
- How can you validate the ports detected as being open; that is, how can you compare this with a target state? Besides an initial scan with subsequent manual checks, you might be able to retrieve a target state documented in an asset database or a configuration management system.

Figure 2 shows a somewhat simplified view of a typical scenario for using Dr. Portscan. In this scenario, an instance of Dr. Portscan is placed inside an HEI network. It receives the scanning results of four different scanning tools, which are partially placed outside the own HEI network, i.e., one scanner is placed inside the network of another HEI (the *hosting provider*) and the other scanner is

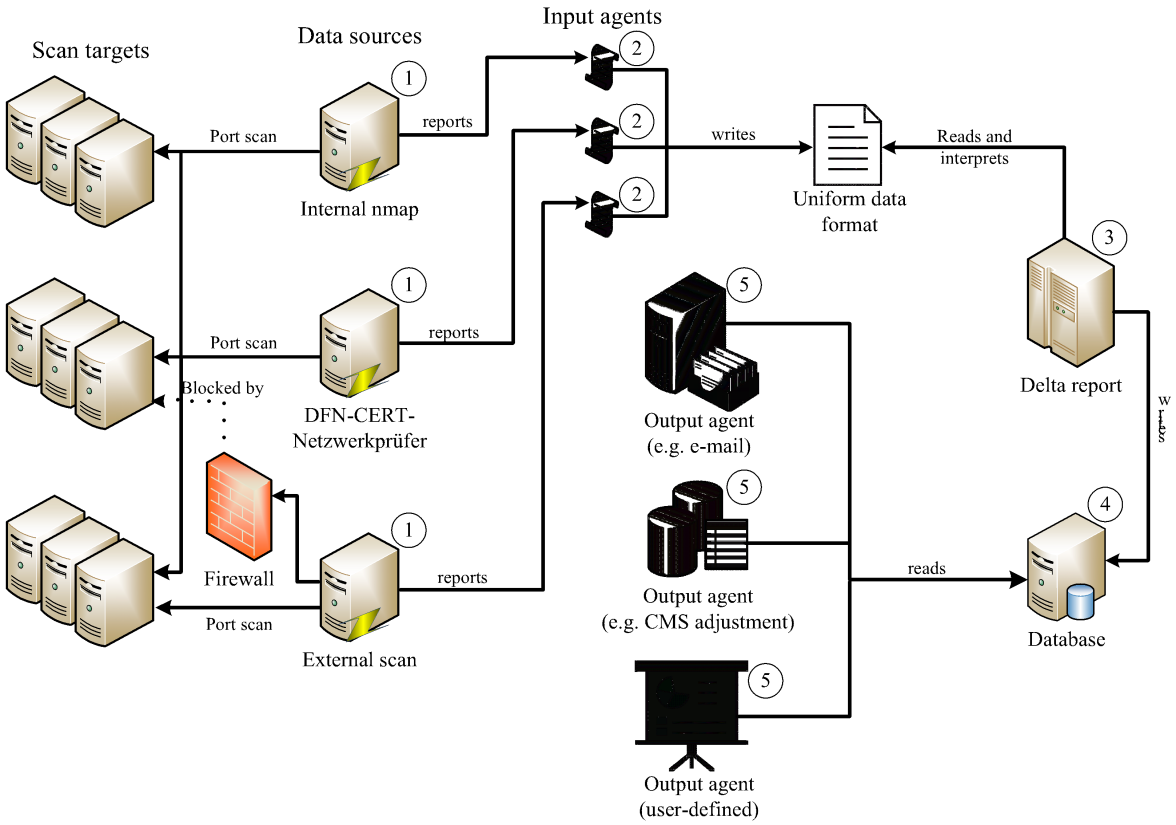


Figure 3: Dr. Portscan system architecture

maintained by the German DFN, the German national research and education network (called the *DFN Netzwerkprüfer*). The Netzwerkprüfer is placed outside the network and runs every 28 days with a predefined set of parameters. The scanner at the hosing provider is configured by us and so it can scan our network as we need. The remaining two sensors are inside the own network. There is a normal nmap scanner and as a proof of concept a samhain scanner, which demonstrates that Dr. Portscan is not limited to the nmap port scanner.

This is the intended application scenario in which two or more HEI cooperate to serve a scanning instance with other HEIs.

4. SYSTEM ARCHITECTURE OF DR. PORTSCAN

The scripts on which Dr. Portscan is based compare an arbitrary number of port scan result lists created by different computers. This distinguishes Dr. Portscan from the popular Ndiff tool, which is only able to compare exactly two nmap scan results.

An overview of the system architecture of Dr. Portscan is depicted in Figure 3. In the following, its most important components are:

- Data sources (1) can be any port scanning tools.
- As all these port scanners have their own data format and outputs, there is a need for agents (2), which are responsible for converting the product-specific scan results into a uniform format. The input agent for nmap, for example, contains parsers for XML and grepable nmap output as of version 5. The overhead for implementing additional input agents depends on the port scanning tool, which should be integrated, but it is typically reasonable because the code segments are needed to modify simply consist of a list of the detected IP addresses and open ports.

Table 1: Change Types

Type	Description
0	No change compared with last scan
1	New machine (IP address detected for the first time)
2	Machine no longer running
4	Changed DNS name
8	New open port, not previously open
16	Port now closed (port previously open)
32	Port opened again after being closed in the meantime

- The delta reporter (3) transfers the results delivered by the input agents to a database (4).
- This database stores and evaluates the results of all port scans. The information generated in the process indicates changes compared with recent scans. The database stores all the information for every scanned machine and every port that has drawn attention to itself. The most important thing to watch is the changes compared with the last port scan run for the same entry. Table 1 shows the supported change types.
- Freely definable output agents (5) can provide an overview, create reports about changes on the individual subnets, mail information to the administrator in charge, or trigger arbitrary scripts to retrieve more information about the service hiding behind the newly opened port.

Dr. Portscan is designed to use a centralized relational database, initially to collect the results of the decentralized port scans in a uniform format. Because the input agents, the delta reporter, and the output agents are all implemented in Perl, you can use a wide spectrum of open source and commercial database products. The minimal sample configuration relies on SQLite. This sample configuration makes it easy to try out Dr. Portscan without having to set up a database server and a

new database. For larger installations, it makes sense to use MariaDB, PostgreSQL, or some other relational database management system for which the Perl DBI modules are available.

4.1. Data Sources and Input Agents

The port scanning tools (or data sources, as they are known in Dr. Portscan) are initially configured independently of the input agents. For example, a cronjob could be configured, which runs automatically a nmap instance once a day on a Linux machine to scan all the ports in a network range equivalent to a legacy Class B network:

```
/usr/bin/nmap -vv -oG - -p- 10.1.0.0/16 > /tmp/nmap-results.txt
```

This nmap call scans all port numbers (`-p-`) in the possible range from one to 65535 in the network `10.1.0.0/16` and writes the results to the file `/tmp/nmap-results.txt`. The parameter `-vv` increases the level of details in the result file. Finally, the parameter `-oG` sets the output to a grepable format. Another output format is the XML based nmap output (parameter `-oX`) which is used in general for Dr. Portscan.

The problem for Dr. Portscan is that every port scanning tool has its own output format; even nmap has four fundamentally different output formats. Therefore, it is necessary that input agents are used to convert the results of a data source into the Dr. Portscan format.

Each data source is assigned to an input agent; its main task is to generate a list of all IP address and port pairs detected as open by the data source. The agent reads and parses the output from the scan tool. The preceding call to nmap would return the following text output on a web server (e.g., `www.lrz-muenchen.de`):

```
Host: 129.187.254.92 (www.lrz-muenchen.de)      Status: Up
Ports: 80/open/tcp//http///, 443/open/tcp//https///, 8080/open/tcp//http-proxy///, 8443/open/tcp//https-alt///
Ignored State: filtered (65531)
```

The results from the input agent – a simple text file with an IP address, a port, the protocol details, in addition, an optional timestamp in each line – are then either deposited for the delta reporter to pick up or pushed to another system (e.g., using scp). This intermediate step is often necessary because the SQLite database used by Dr. Portscan in the default configuration (without addon software such as cubeSQL) is not accessible on the network and because other databases are often hardened against Internet access by a firewall, thus preventing direct entry of the results in a central database.

4.2. Delta Reporter, Output Agents, and Reports

The delta reporter first transfers the port scan results returned by the input agents to the central database. The reporter checks for each IP address and port pair, taking into consideration the Layer 4 protocol, whether an entry for the current scanner already exists, or whether a new entry needs to be created. If a new entry is required because the IP address and port pair is not known yet, then the new entry gets the change type 8 (new opened port).

More interesting is the case if existing entries have to be updated because the IP address and port pair has been seen before. Then the delta reporter compares the actual scanning result with the previous runs of the scanner. This analysis now leads to one of the in Table 1 presented status results. Then, the appropriate change type is newly set in the database entry of the IP address and port pair. An important note is that port scanners and even the import agents does only know open ports. Closed ports, which are therefore no longer reported in this list, are recognize by the delta reporter because of the missing item in the result list.

After all the port scan results have been imported, the output agent can evaluate the central data repository. Because changes compared with the status quo are of primary interest, output agents typically only look for database entries with specific change types. This leads to a result list, where all new ports, machines or DNS entries are mentioned and closed ports or no longer reachable machines, too.

In general, the port scanner runs multiple times a day, e.g., every four hours, to detect changes and the report is only generated once a day. So, it is unlikely that one event is reported more than once. But in the case that multiple reports are generated for different groups, it could be that one group would like to have their results once a day and the other group only once a week. To prevent an output agent from processing the same entry multiple times, the agent enters its own identifier in the corresponding column of the database table.

The output agents in the preconfigured sources are primarily written for an email notification, which is sent in regular time intervals to the responsible administrators. Because of the modular development of Dr. Portscan it is easily possible to write other output agents, for example agents, which takes the result list to feed it into a vulnerability scanner. This would help to detect vulnerable services at an early stage before an attacker discover them. Another possible output agent is a tool, which checks if the changes are documented in a central database, e.g., a ISO/IEC 20000 CMDB (configuration management database). This helps an organization to have every time an up to date documentation of their network and systems.

Dr. Portscan offers a preconfigured output agent that gives you an overview of all changes since the last report, but you can restrict evaluation by the output agents in a targeted way. For example, you might just want to see the results for specific subnets or specific events (e.g., newly opened ports) to compare these results with a defined target state.

4.3. Optimizations

In some situations the port scanners do not deliver deterministic results, so the results could be corrupted. This could be, when the scanner is outside the own network and there are problems in the network connection, the port scanner is temporary damaged (e.g., a system restart on the scanner side) or the target system is temporary unreachable. This would lead to two reports where the first report says “port closed” and the second “port opened”, while on system side no change is executed. To avoid this, ports are first reported to be open or closed when freely configurable scanner-specific thresholds are exceeded. These thresholds contribute toward stabilizing the port scan and help you avoid false alarms.

Discrepancies between the results of various scanners are typically legitimate and can be caused by different scan times, scan scopes, or scanner locations. These different views of the same system need to be processed adequately in the report, for example, by listing the current results for the different scanners each time the report is generated. The configuration assigns each scanner to a group, within which all sensors should have the same view to be able to detect inconsistencies reliably.

Changes to the scanner infrastructure can make it necessary to clean up the stored data from time to time. Optional scripts in the Dr. Portscan distribution can delete entries with very old timestamps, results from a specific scanner (if you are discontinuing its use), all entries for a specific network range, or all entries for an IP address (e.g., if you are replacing one machine with another that will use the same address).

5. USING DR. PORTSCAN

5.1. Installation and Commissioning

The following installation description assumes that the delta reporter, the output agents, and the database are installed on the same machine. The scanners may run on different systems as long as they are capable of transferring their scan results to the central delta-reporting instance. The current version of Dr. Portscan (Hommel, Metzger, & Eye, 2014) is available from a Git repository. As an alternative to downloading from your web browser, you can retrieve the complete repository as follows:

```
git clone git://git.lrz.de/DrPortScan.git
```

The preconditions for running the simplest installation variant are SQLite3, Perl, and the following Perl modules, which you can install via CPAN along with their dependencies: DBD::SQLite, XML:: LibXML, DateTime, and DateTime:: Format:: Strptime. To launch the install, the setup.pl script has to be called, which checks to see whether the required Perl modules are in place. If the modules are

not in place, an error message is displayed stating which modules is still needed to install. Additionally, the script handles the task of creating the directory structure for the input and output files, which is also necessary for running Dr. Portscan. To create and initialize a SQLite database, the script create_db.sh creates all necessary tables and entries. After creating the database, the script also registers a number of test scanners, which can be used as templates for the own scanner definitions. If you prefer not to use this option, you can manage the scanners later on using the configuration.pl script. In addition to listing the scanners currently registered in the database, this script also lets you enter new scanners and modify or remove existing scanners.

5.2. Scanning

Consider the following example, which relies on the standard scanning tool nmap. Dr. Portscan already comes with a preconfigured nmap input agent based on the XML based output. For an nmap scan with detailed XML output, use the following command:

```
nmap -oX /<path>/<to>/<file>/nmap-xml_scanner_timestamp.xml <IPrange>
```

If Dr. Portscan does not provide a matching input agent, you need to develop an agent for your own scanner software; you can use the existing template as the basis for your development. You need to transfer the scan output to the Dr. Portscan input folder on the central machine. The output filename must follow a specific pattern to help identify the input agent responsible for processing the file. You also need to state which scanner the file came from and when the scan was performed. The scanner ID is the same as the ID used to register the scanner with the centralized database, and the date must use the YYYYMMDDHHMMSS format; thus, the filename looks like this: <input-agent>_<scanner>_<date>.*. It isn't important how the files make their way from the external scanners to the central delta reporting instance. If you want to restrict access to the delta reporting system, you can use scp or sftp as cronjobs to retrieve new scan results.

5.3. Delta Reporting

The central component in Dr. Portscan is the delta reporting instance. The input_watcher.pl script checks to see whether new scan results are available for processing. The files are first sorted chronologically; then, the script finds the matching input agent, calls the agent to convert the files to a uniform data format, and sends the results to the delta reporter for ongoing processing. If this processing completes without error, the file is moved to the old directory; otherwise, it is moved to failed. (To make sure the input watcher script is run at regular intervals, create a cronjob.) The delta reporter now compares the current results with those of the previous scan and enters the results in the database. The output agents then modify this output for further use. A first step in a typical application is the xml_out.pl output agent, which outputs the detected changes as an XML document. You can then convert the document to a plain-text version using the xml2plaintex.pl script and mail the results as needed. Alternatively, you can convert the XML document to HTML and view the results in a browser.

6. THE FUTURE OF DR. PORTSCAN

Dr. Portscan is actively being developed and improved; the next version will include a multitenant-capable configuration and reporting tool based on a web service. This tool will allow administrators who are responsible for part of a network to define the scope of the scan and the boundaries of the report in a web front end. Additionally, reporting will be improved for IPv4/ IPv6 dual-stack environments to help identify inconsistent IPv4 and IPv6 firewall rules. In addition, the developers are working on an output agent for reporting port scan results to a security management system such as Alien-Vault OSSIM or log management solutions as Splunk.

Additionally, we plan an interface for the integration of netflow analysis. Netflows are generated by routers or L3 switches and give an overview about the traffic, which enters or exits the network component. The including information involve the source and destination of the flow, the used ports and the amount of data send in this flow. This information can give a hint, if a system communicate with another system, but right by UDP traffic it is not certain, that the destination machine is listening. However, we plan to use netflows to increase the analysis capability of Dr. Portscan, because there are ports in the wild, which does not listen in the normal way. These ports are only

open and listening if a special port knocking sequence is performed. A normal port scanner does not know this sequence and so it only sees a closed port. But if there is an huge amount of traffic to this closed port, it could be that this is a hidden port. With a little analysis of the netflows it is possible to extract the port knocking sequence and to perform this at the next port scanning.

7. CONCLUSION

Port-scanning one's own campus network is one technical measure how HEIs can keep track of their network service assets and associated risks. However, the output of port scan tools like nmap quickly becomes very unclear when several hundreds of servers are analyzed. Dr. Portscan helps to assemble a big picture by combining results from an arbitrary number of port scan runs distributed in space and time, i.e., port scans are run from several locations, such as from within as well as outside the HEI's own research network, in arbitrary time intervals. It also assists with the analysis of these consolidated results by creating the name-giving delta-reports, so campus IT security team can focus on the in-depth analysis of changes to the IT infrastructure.

After describing the motivation for and use cases of Dr. Portscan, we presented the basic points of how HEI administrators should set up their own Dr. Portscan infrastructure along with port scanning tools. We then presented the inner workings of Dr. Portscan in detail by discussing its overall architecture and the most important workflows. Finally, we've shown how Dr. Portscan can be used in practice and gave a short outlook to the next steps of its ongoing development.

Dr. Portscan is open source software and can be downloaded from https://git.lrz.de/?p=DrPortScan.git;a=tree ; any feedback from users and suggestions for further improvement are highly welcome.

8. REFERENCES

Hoffmann, M. (21. 02 2014). Von SASER-SIEGFRIED Project - Project Information: http://www.celtic-initiative.org/Projects/Celtic-Plus-Projects/2011/SASER/SASER-b-Siegfried/saser-b-default.asp

Hommel, W., Metzger, S., & Eye, F. v. (20. 02 2014). Dr. Portscan Git Repository: https://git.lrz.de/?p=DrPortScan.git

Lyon, G. (2008). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, USA: Insecure.Com.

9. AUTHORS' BIOGRAPHIES

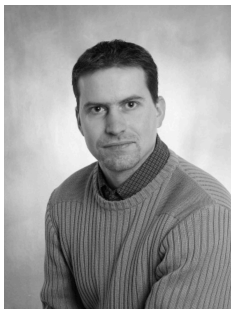
Felix von Eye is member of the communication networks planning group at the Leibniz Supercomputing Centre and works in different research projects in the area of network management, IT security, and intrusion detection.

He studied computer science and mathematics and attained in 2009 a diploma degree in mathematics from Augsburg University. Currently he is working on his PhD in dynamic and automatic threshold methods for intrusion detection systems. At the Ludwig Maximilians University Munich and the Technical University Munich he supervises different thesis and is involved in the security lectures.

Wolfgang Hommel is the Chief Information Security Officer of the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities, where he is also the head of the communication networks planning group.

He studied computer science at Technische Universität München and has a Ph.D. as well as a postdoctoral lecture qualification from Ludwig-Maximilians-Universität in Munich, Germany, where he teaches information security lectures and labs. His research, for which he was granted the Karl Thiemig foundation's young academics award in 2011, focuses on information security and IT service management in complex large-scale and inter-organizational

scenarios.



Stefan Metzger is member of the communication networks planning group at the Leibniz Supercomputing Centre. He holds an international CISSP certification. As head of the LRZ security working group his main focus lays on ISO/IEC 27000-based security management.

He attained in 2005 a Diploma degree in Computer Science from Technical University Munich. Currently he's doing his PhD in security management in large-scaled, inter-organizational infrastructures and offers lab courses in security at Ludwig Maximilians University (LMU) Munich.

Background information

The Leibniz Supercomputing Centre (LRZ) is the common IT service provider of the universities and higher education institutions in the greater Munich area, Germany. It offers IT services for more than 130,000 regional users and operates the Munich Scientific Network, which is part of the German national research and education network. LRZ is also one of Germany's national supercomputing centres and offers it high performance computing services to scientific communities across Europe.

Acknowledgment

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.