Enhancing security testing via automated replication of IT-asset topologies

Henk Birkholz, Ingo Sieverdingbeck, Nicolai Kuntze, Carsten Rudolph
Fraunhofer Institute for Secure Information Technology (SIT)
Darmstadt, Germany
{henk.birkholz|ingo.sieverdingbeck|nicolai.kuntze|carsten.rudolph}@sit.fraunhofer.de

***Abstract*—Security testing of IT-infrastructure in a production environment can have a negative impact on business processes supported by IT-assets. A testbed can be used to provide an alternate testing environment in order to mitigate this impact. Unfortunately, for small and medium enterprises, maintaining a physical testbed and its consistency with the production environment is a cost-intensive task. In this paper, we present the Infrastructure Replication Process (IRP) and a corresponding Topology Editor, to provide a cost-efficient method that makes security testing in small and medium enterprises more feasible. We utilize a virtual environment as a testbed and provide a structured approach that takes into account the differences between a physical and a virtual environment. Open standards, such as SCAP, OVAL or XCCDF, and the utilization the Interconnected-asset Ontology—IO—support the integration of the IRP into existing (automated) processes. We use the implementation of a prototype to present a proof-of-concept that shows how typical challenges regarding security testing can be successfully mitigated via the IRP.**

I. Introduction

Security testing is a common procedure to evaluate IT-security compliance in small and medium enterprises (SMEs). It requires dedicated resources and involves considerable costs. Hence, it is a priority for SMEs to implement efficient security testing processes to maintain a required level of IT-security while reducing costs. Arkin et al. highlight that appropriate tools can help to reduce security testing costs, but these tools are usually not applicable at the design level [1]. In this paper, we present an Infrastructure Replication Process (IRP) that enables SMEs to conduct security testing in a virtual environment (VE) instead of the physical environment (PE), thereby mitigating the disadvantages of security testing in a production environment. The IRP

1) acquires a snapshot of an existing computer network (the IT-asset topology) and its participants (hosts and network components) and
2) replicates this snapshot in the VE.

In this paper, the term *replication* is applied to the task of creating virtual machines based on an existing physical hosts or—analogously—creating virtual network components based on physical network components. The IRP uses common off-the-shelf virtualization methods to create a virtual testbed that contains the VE. IT-asset topologies acquired via the IRP can be modified before replication. This intermediary process step, supported by an interactive Topology Editor, enables SMEs to evaluate changes to the network design in the VE before applying them in the PE (basic quality assurance). The goal of our work is to reduce the cost of security testing for SMEs by aggregating related tasks into one canonical process (the IRP) and to provide a single interface for process control (the Topology Editor).

The remainder of this paper is structured as follows. First, related work in the context of security testing is presented. The focus lies on best practices in security testing, security automation, formal representation of network topologies and common virtualization methods. Section II highlights the main differences between testing in a physical environment and testing in a virtual environment. In section III common challenges of security testing regarding SMEs are summarized. Section IV presents a description of the Infrastructure Replication Process (IRP), its basic operations and involved components. The Topology Editor is introduced and its relationship with basic functions of the Interconnected-asset Ontology [2] is discussed. Section V presents a proof-of-concept and corresponding evaluation based on an artifact-building approach, and section VI concludes this paper with implications for future work.

II. Related Work

The National Institute for Standards and Technologies (NIST) provides a detailed guideline for information security testing [3]. This includes a number of security testing techniques that can be applied to evaluate the behavior of interconnected IT-assets. Examples are, *Target Identification and Analysis Techniques*, such as network discovery and vulnerability scanning, or *Review Techniques*, such as system configuration reviews and network sniffing. The guideline also refers to incorporating existing asset inventories and conducting a walkthrough of a facility as preliminary discovery processes that are part of security testing. These manual process steps are a mandatory prerequisite for effective security testing in order to identify IT assets that were not found by technical means. Implemented security measures can inhibit discovery mechanisms inherently due to their security goals [4]. Therefore, the IRP presented in this paper requires an initial list of IT-assets that represents the hosts and managed network components to be included in the replication process. This list is aggregated by manual post-processing of automatically acquired asset information.

Information security tests can be automated. The Security Content Automation Protocol (SCAP [5]) that is developed in a cooperation of NIST and the Massachusetts Institute of Technology Research Establishment (MITRE [6]) provides structured representation for identified assets (SCAP-AI), tests and automated checklist definitions. In the context of SCAP, the Open Vulnerability Assessment Language (OVAL) can be used to map high-level technical checks to low-level details of executing those checks [7]. Despite its given name, the definitions in OVAL are not only able to test for vulnerabilities but also to define tests to identify asset characteristics, e.g. a specific operation system running on a host. In OVAL *definitions*, test definitions are used in individual *criterion* elements that are aggregated in the *criteria* definition. Each *criterion* element includes an URI pointing to an explicit test procedure (identified by a distinct ID) that is available through the OVAL repository [8]. OVAL Test ID 7914 (oval:org.mitre.oval:tst:7914), for example, contains a regular expression that will match if an operating-system-specific system file is parsed. Therefore, vulnerability testing is a combination of testing for prerequisites (system characteristics) and actual vulnerabilities (software version or behavior). In the IRP, custom *criterion* elements are used to evaluate the support of given features in the VE.

SCAP also employs the eXtensible Configuration Checklist Description Format (XCCDF [9]). In this format, checklist items can be associated with checks, such as OVAL definitions, and preconditions, such as preliminary checks that are represented by the <xccdf:require> element. A common example for an *require* element highlighted in the XCCDF specification [9] is ”<xccdf:requires idref="xccdf_org.example_rule_passwd-exists"/>”. This kind of preliminary requirement is found in checklist items to evaluate certain file system privileges. In the IRP, *require* elements identify security tests that can be conducted only in the PE due to conceptual discrepancies between PE and VE.

In the context of the VISA project[1], security tests defined via OVAL are managed by the Control and Management Framework (OMF). OMF is a control, measurement, and management-framework, specially designed for application in testbeds [10] with a strong focus on aggregating and summarizing complex measurement results. OMF supports complex interdependent test scenarios that can be adapted to different IT-asset topologies. Complex OMF test results can also be broken down to produce a result state required by OVAL state definitions.

Security testing often requires access to host operation systems and the installation of test software. For example, file integrity checker or system configuration review frameworks are a common part of security testing [3] that requires the installation of appropriate tools. Hence, the installation of test tools with OMF capabilities to conduct thorough security testing complies with the best practice described by the NIST security testing guideline.

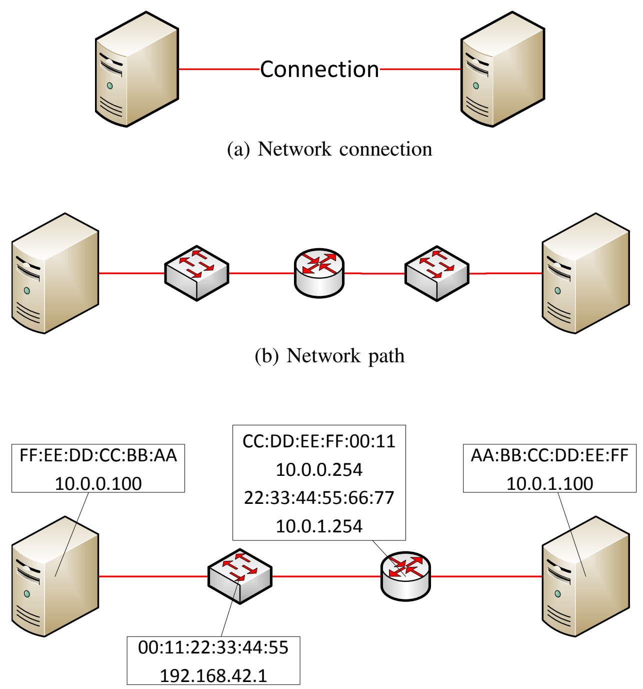
To create a virtual replica of existing IT-infrastructure the virtualization methods employed have to mimic the functionality and characteristics of the original physical IT-assets. The VISA project is focused on enhancing security testing. Hence, the virtual environment must not interfere with the security tests conducted. Unfortunately, the major issue with virtualization techniques is performance [11]. Especially performance behavior between virtualized entities inside a VE significantly deviates from the performance behavior of a PE [12]. As a result, performance benchmarks—in the context of stress tests and distributed denial of service (DDoS) test scenarios—cannot be migrated into a VE today. There are also specific features a VE cannot provide. The openvswitch project [13], for example, provides an extensive list of features and capabilities on layer 2 [14], but Multiple VLAN Registration Protocol (MVRP), is not one of them. MVRP is a good example of a technical measure that can reduce costs due to reduced configuration overhead, but that can also introduce inconsistent system states that can be exploited. If the VE cannot provide a feature, such as MVRP, corresponding security tests regarding this feature also have to be excluded.

A formal representation to acquire, store, modify and provide snapshots of network topologies is required for the IRP. The Interconnected-asset Ontology used by the IO tool-set [2] can acquire and process complex topological configuration and state information from heterogeneous IT-assets in high detail. Furthermore, the acquisition procedures of the IO tool-set can be fully automated. This is an important prerequisite for replicating existing infrastructure in SMEs: managed network components deployed by SMEs are often heterogeneous and processing of every configuration detail manually is not feasible. Most importantly, the IO tool-set focuses on automated acquisition and provision procedures, which enables integration into the automated IRP. While the ontological representation is stored in the W3C's Web Ontology Language (OWL [15]) format, the IRP utilizes the less complex Resource Description Framework (RDF [16]) metadata data model that can be retrieved with the IO tool-set via customizable retrieval procedures.

The utilization of a virtual environment as a testbed is focus of various publications. Arnes et al. present in their work the ViSe project [17]. It's goal is to improve the evaluation of intrusion detection systems. Preconfigured basis systems that contain 10 versions of popular operating systems, and 40 exploits against them provide the foundation of the ViSe testbed. To support testing, instanced reference systems can be assigned specific roles in the testbed, such as, *Attacker*, *Detector* or *Victim*. Similar to the ViSe project, additional hosts with specific roles in regard to security testing can be introduced into the VE during the IRP presented in this paper.

With the Cyber DEfense Technology Experimental Research testbed (DETER), presented by Benzel et al., it is possible to create a new IT-asset topology by using cascading installation scripts [18]. Experiments in DETER are conducted with the Security Experimenters' Workbench (SEW)

[1] http://www.visa-project.de

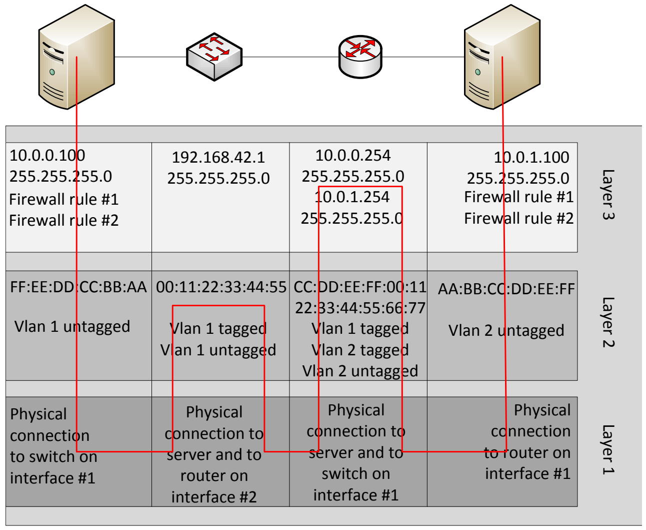


(a) Network connection

(b) Network path

(c) Detailed network path

Fig. 1.



10.0.0.100 255.255.255.0 Firewall rule #1 Firewall rule #2	192.168.42.1 255.255.255.0	10.0.0.254 255.255.255.0 10.0.1.254 255.255.255.0	10.0.1.100 255.255.255.0 Firewall rule #1 Firewall rule #2	Layer 3
FF:EE:DD:CC:BB:AA Vlan 1 untagged	00:11:22:33:44:55 Vlan 1 tagged Vlan 1 untagged	CC:DD:EE:FF:00:11 22:33:44:55:66:77 Vlan 1 tagged Vlan 2 tagged Vlan 2 untagged	AA:BB:CC:DD:EE:FF Vlan 2 untagged	Layer 2
Physical connection to switch on interface #1	Physical connection to server and to router on interface #2	Physical connection to server and to switch on interface #1	Physical connection to router on interface #1	Layer 1

Fig. 2. Property chain representing multiple *connected-To* relationships in IO

that can be used via a specialized GUI component called the Experiment Specification and Visualization Tool (ESVT). The DETER testbed is a shared infrastructure designed for medium-scale repeatable experiments in computer security, and it has a strong focus on malware analysis. The Topology Editor presented in this paper is similar to the ESVT, but also incorporates representing and processing IT-asset topology information acquired from heterogeneous IT-assets.

The Lincoln Laboratory SIMulator (LLSIM) is an easily configurable network simulator that can produce a wide variety of data sets without expensive testbeds [19]. It achieves high scalability and faster-than-real-time network performance by omitting ”complex interaction between testbed components” and thus avoiding ”detailed packet-level and flow-level simulation”. Nodes are represented by virtual machines in a manner that allows the simulator to mimic the behavior of a complete IT-asset topology. The main technical difference between the LLSIM and the IRP is the complexity regarding the formal representation of interconnected IT-assets. To enable reliable and deterministic security testing, the re-creation (replication) of networking mechanics in high detail is mandatory.

III. Security Testing in SME

Traditional security testing in the production environment of SMEs involves typical challenges and drawbacks:

1) Security testing requires up-to-date documentation about the IT-infrastructure, its configuration and also its typical state while in production. Supported business processes and corresponding security guidelines also have to be documented. Unfortunately, documentation is often out-of-date, incomplete or inconsistent [20] which has a negative impact on security management related processes.
2) Security testing that includes invasive procedures such as penetration testing or the deployment of evaluation software in the production environment can have a negative impact on security goals such as availability and integrity of supported business processes [21].
3) The time required to deploy and configure testing-equipment in the corresponding IT-infrastructure is often extensive. For example, manual configuration of equipment or adaptation to corresponding interfaces (appropriate physical ports, networks taps and VLAN, QoS, or subnet configuration) can be necessary for each new testing procedure, increasing costs. Sanchez et al. show that this kind of complex management and costly maintenance can make security tools inadequate for SME [22].
4) Enforced security policies have to be taken into account when trying to acquire raw asset information from production IT-assets in the PE via physical or remote access. Security policies can inhibit this preliminary acquisition of asset information (e.g. acquisition of the current state of dynamic device configuration) or compromise the results of security testing conducted in the VE (e.g. IPS (Intrusion Prevention Systems) policies can interfere with IDS auditing). Unfortunately, adapting existing security policies to enable specific auditing test scenarios can also introduce new vulnerabilities as a temporal side-effect.

IV. Security Testing supported by the IRP

In order to conduct security testing in a VE, it is paramount not to compromise testing results due to discrepancies introduced by virtualization mechanisms. A virtual environment

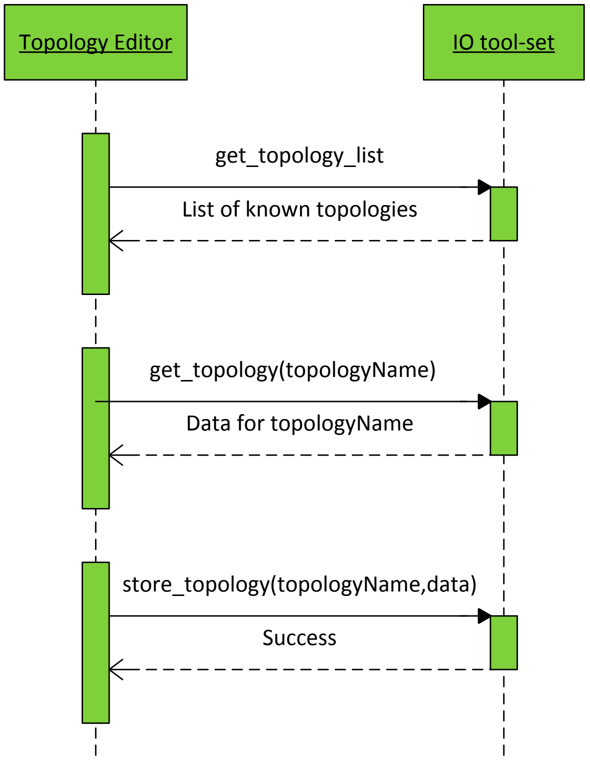


Fig. 3. Sequence diagram illustrating the *get* and *store* operations

differs from a physical environment. The most important difference is service behavior under stress [12]. While evaluation of relative behavior under stress can still be feasible, e.g. quality of service classes that police certain throughput in relation to achievable bandwidth can still show a correct behavior, absolute behavior is distorted by the introduction of abstraction layers and more complex distribution of CPU time. For example, a virtual managed network component's maximum bandwidth can be lower than the maximum bandwidth of a physical one due to the missing optimization of specialized hardware. In the context of the IRP, this is called a *conceptual discrepancy*.

There are also discrepancies in the capabilities of a VE compared to a PE. Software implementations of virtual switches often cannot keep up with features provided by hardware network components. For example, open-vswitch does not support the Multiple VLAN Registration Protocol (MVRP) which is often a focus of an security audit. In the context of the IRP, this is called a *feature discrepancy*.

Security tests that cannot be conducted in a VE due to *conceptual* or *feature discrepancies* have to be omitted and documented, respectively. Every discrepancy that can be identified is taken into account by the IRP. At this point, our IRP relies on a manually composed, machine-readable *discrepancy list* that can be processed by an OVAL definition automatically. Every security test to be conducted in the VE has to be provided in the Extensible Configuration Checklist Description Format (XCCDF [9]). Each checklist item of this *security test checklist* refers to a security test represented by an OVAL *definition* element.

If there is a *conceptual discrepancy* associated with the security test an XCCDF *require* element is used in the security test checklist that refers to an OVAL *definition* processing the discrepancy list. The resulting state of the target of the OVAL definition then exhibits the corresponding discrepancy, the required state is not met and the actual security test is not conducted.

If there is a *feature discrepancy* associated with the security test an additional OVAL *criterion* element is added to the corresponding OVAL definition of the security test. This is a fallback procedure. In general, a check defined via an OVAL *definition* should be able to detect the absence or inappropriateness of a feature on the involved target host. Unfortunately, virtualization mechanisms hide features from guest hosts. If an OVAL definition cannot assess the required state necessary to produce a valid result, the additional *criterion* element has to be added to omit the security test.

V. IRP STRUCTURE

The IRP presented in this paper takes into account the characteristics of virtualization techniques described in section IV and mitigates the drawbacks of security testing in a production environment highlighted in section III. The IRP can be divided into two related domains:

1) replication of the interconnected IT-asset topology (network replication), and
2) replication of distinct hosts participating in the network (host replication)

A. Host Replication

Applicable methods to virtualize existing hosts (and their operation system and services, respectively) are strongly dependent on the type of operation system and available interfaces. On the one hand, modern host deployment and management systems, such as puppet[2] and the Logical Volume Manager (LVM) or Windows Management Instrumentation (WMI) and the corresponding creation of Virtual Hard Disk (VHD) images, enable an automatic acquisition of disk images that can be used to operate a native installation in a VE. On the other hand, disk images derived from older operation systems have to be acquired manually and running them in a VE can require considerable manual adaption. If a disk image is acquired and adapted, it has to be virtualized via any common virtualization method, such as, VMWare/vSphere, KVM/openstack. The replication of hosts is primarily a technical challenge that requires resources dependent on the type of operating system.

B. Network Replication

Replication of an existing network topology requires a formal representation of interconnected IT-assets and a virtualization method to implement the characteristics of the original

[2] https://puppetlabs.com/

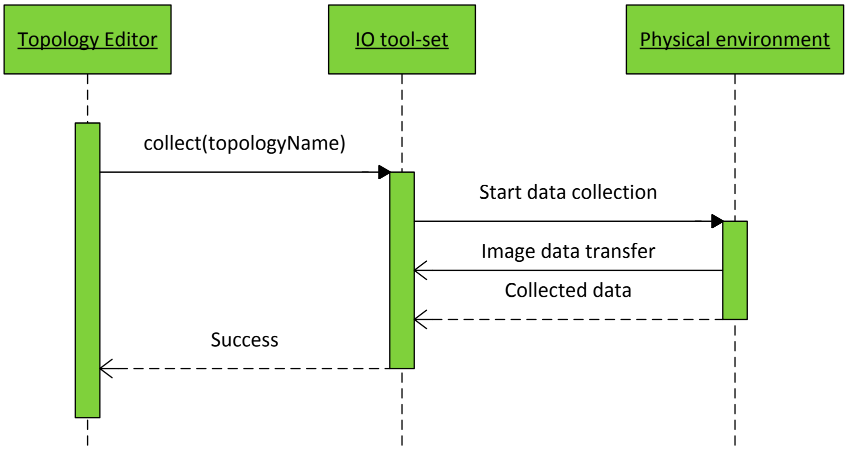


Fig. 4. Sequence diagram illustrating the *collect* operation

network topology in the VE. The formal representation used by the IRP is provided by the IO tool-set [2]. It utilizes the interconnected-asset ontology and can store asset information, such as configuration, state and dynamic relationships between network components, in high detail. Raw asset information is acquired from managed network components, via transports such as, SNMP, SSH or SOAP, automatically. It is then stored in the IO. Each snapshot of the network topology stored in the IO can be saved and restored individually. The command protocol IO-X (IO eXchange protocol) developed by the VISA project provides the means to create, transfer/modify and delete snapshots as part of the IRP.

C. IRP components and operations

Four components interact via the IRP:

1) Interconnected-asset Ontology (IO)
2) Topology Editor (TE)
3) Physical (Source) Environment (PE)
4) Virtual (Destination) Environment (VE)

To simplify the interaction of the IO tool-set with other architectures, there are four core operations that can be customized for further applications.

1) The IO tool-set: The IO tool-set functions as the central repository regarding data about network topologies and corresponding hosts, including their disk images. The Interconnected-asset Ontology stores every information necessary to create a working replica of a PE in a VE. The current version of the ontological T-Box, developed by the VISA project and used by the IO tool-set to represent a PE, is published online [23]. The IO tool-set stores the ontological representation in OWL format [15]. Figure 2 shows the level of detail compared to

1) Figure 1a—a simplified connection concept, i.e. representing a connection. This concept is often found in formal top-level representations, e.g., the asset ontology presented by Aime et al. [24],
2) Figure 1b—a network path concept composed of individual hops. This representation is often used in network maps that are employed by, e.g., nmap (zenmap topology [25]) or nagios (nagios map [26]),
3) and Figure 1c—an enriched network path that aggregates information about distinguishable network layers and corresponding addresses for each hop. This representation can be found in sophisticated network management systems, such as the HP Network Management Center [27], or IBM Tivoli [28].

The representation in the IO is strongly influenced by the common OSI layer representation. While this influence is also found in other conceptual models, such as the DMTF Common Information Model [29], using a level of detail as highlighted in Figure 2 an actual implementation consistently, is a novel approach. To desist from this level of detail is typically justified by the large amount of aggregated data that has to be processed. In large scales, manual aggregation of raw asset information is conceptually not feasible and automatic acquisition thus made mandatory. The complex ontological representation (OWL/XML) that is used in the IO can be easily broken down into less complex representations, such as triplets in RDF/XML format [16] or tables (SQL, CSV, etc.). On the one hand, this allows for application specific selection of an appropriate level of detail that is exchanged with a consumer of information. On the other hand, fast and efficient triple stores, e.g. AllegroGraph [30], can be utilized in order to keep high-detail operating performance on a level that is necessary for application in SMEs despite the large amount of data. The IO tool-set has proven to be fast in large scale application [2], representing over 2,000 managed network components and over 10k associated host endpoints.

On protocol level, the IO tool-set provides four primary IRP operations named *collect*, *replicate*, *get* and *store*. They are implemented in the IO eXchange protocol (IO-X) and are presented as sequence diagrams in Figure 3, 4 and 5.

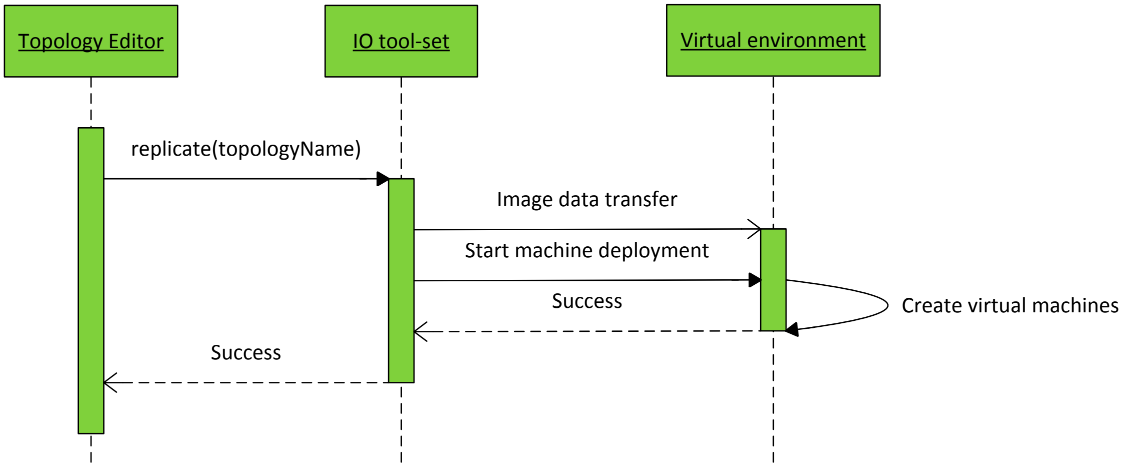


Fig. 5. Sequence diagram illustrating the *replicate* operation

The *collect* operation initiates an IO *acquisition procedure* [2] and creates a new formal ontological representation of the network topology found in the PE. Existing policies in the PE (as described in section III item 4) have to be aligned with the *acquisition procedures* of the IO tool-set (in the PE) in order to assemble a complete interconnected IT-asset ontology. The *replicate* operation complements the *collect* operation and translates the interconnected IT-asset information into an appropriate format that can be operationalized by the employed virtualization method on operation system level. For example, the configuration of bridge interfaces including a VLAN id layout can be provided as parameters that are required to execute corresponding system configuration tools in the hosting systems of the VE. The *get* and *store* operations handle the exchange of information between the IO tool-set and the Topology Editor. Typically, existing snapshots of a network topology are modified in the Topology Editor between *get* and *store* operations. Multiple modified topology snapshots can be stored and replicated via the IO tool-set. Once customized appropriately, the automatic *acquisition procedures* of the IO tool-set enable the creation of up-to-date snapshots of the complete PE, automatically.

2) The Topology Editor: The Topology Editor[3] [31] provides the general user interface required for interaction with the IRP. The back-end communication of the Topology Editor is handled by IO-X. Primarily, there are the five core functions provided by the Topology Editor. These functions are in close relation with corresponding IO-X operations presented in section V-C1:

1) visualization of topologies stored in the IO (IO-X *get*),
2) modification of visualized topologies and configuration (user interaction),
3) storing modified topologies (IO-X *store*),
4) triggering the acquisition of a snapshot of the PE to be represented in a topology (IO-X *collect*),

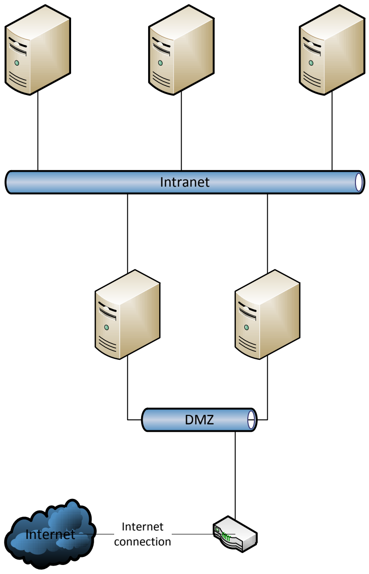


Fig. 6. Acquired IT-asset topology snapshot

5) and replication of a or stored topology (IO-X *replicate*).

In essence, the Topology Editor (TE) is used to visualize and modify snapshots of IT-asset topologies made via the IO tool-set. New IT-assets can be added to visualized topologies in the editor. These added hosts can be customized individually or even be provided by an external source, such as security auditors. The Topology Editor includes a *reference host* for multi-purpose security testing that can be automatically integrated into layer 2 broadcast domains or layer 3 subnets. Figure 6 presents a simple excerpt of a network topology that is processed by the Topology Editor. Figure 7 shows the result of a topology modified by the Topology Editor. Both examples can be created and operated in the VE.

[3] the software can be obtained from Decoit (http://decoit.de/)

The *reference hosts* that have been added as virtual machines via the Topology Editor (Figure 7) can take on different roles. They can be used as *attacker* hosts in the context of penetration testing. The VMs can be configured as *detector* hosts to enable frame and packet sniffing in specific areas of the VE topology. Resulting data is transported via a ”side-channel” network connection that is independent from the rest of the VE topology. This mechanic allows for automatic transport of measurement results without the adaption of security policies, thereby enhancing the quality of the related security tests. The *reference host* can also take on the role of a *management host* to coordinate complex OMF experiments and to present preliminary testing results via the integrated OMF GUI. OMF provides a standard test experiment in the form of an iperf [32] equivalent that is pre-installed on the basic *reference host*. This standard experiment supports the typical connectivity-test procedures that are a common first step during security testing. To simplify the deployment of testing software the *reference host* can be used as a *deployment host* offering installation packages of testing software in various formats (tarballs, portable executables or via packet managers). Each host in the VE can also be offered an emulated USB storage device or a network share to make testing software available. Direct configuration of hosts in the VE (as a fallback) is enabled by off-the-shelf RDP support.

The support of typical features that are highlighted by best-practice studies reduces the overall resource requirements in SMEs significantly. The Topology Editor is intended as an artifact to enable further evaluation in production environments. It is possible to evaluate the effects of modifications to the infrastructure iteratively with the use of the Topology Editor. Customized *reference hosts* that execute tests according to XCCDF checklists and OMF experiments can be placed in the VE to test security policy behavior. If security tests start to fail due to changes in configuration or topological layout these changes can be taken under revision.

D. Physical and Virtual Environment

Interaction with the physical environment and the virtual environment is handled solely by the IO tool-set. Figure 4 shows the PE in the role of a provider of information (in relation to the Interconnected-asset Ontology). In Figure 5, the VE acts as the consumer of information. At this point of the IRP development, design changes to a VE topology that have been validated by security testing have to be adopted into the PE manually. Hence, the PE is always the producer of information and the VE is always the consumer of information.

Interaction with the PE and VE is always initiated manually via the Topology Editor. The current state of the PE can have an impact on the consistency of acquired topology snapshots. During daytime and especially during maintenance time-frames an acquirable topology can be incomplete, e.g. neighborhood relationships can be missing due to temporarily disconnected cabling. This problem is mitigated by allowing only manually initiated *acquisition procedures* (IO-X *collect*) that require a preliminary survey of the PE to confirm its consistency.

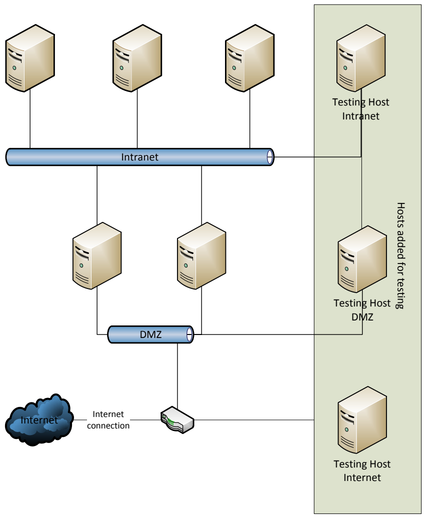


Fig. 7. Acquired IT-asset topology snapshot with additional reference hosts utilized for security testing

VI. Proof of Concept & Evaluation

The evaluation results we present show that

1) the IRP can create a correct replica of a PE including the network topology and participants, and
2) once the VE has been created, security tests can be iterated automatically.

The evaluation of the IRP is based on an artifact-building research approach [33]. As a proof-of-concept we implemented a prototype of the Topology Editor and the IO-X protocol to interact with the IO tool-set. Five IO-query modules [2] have been deployed on hosting systems to operationalize the IO's formal representation in the context of virtualization. IO-query modules produce parameters required by virtualization tools, such as libvirt and libguestfs, to create instances of IT-asset topologies in a VE. Evaluation has been conducted in two evaluation scenarios (ES):

1) replication of a medium sized (~40 hosts) real-world computer network and
2) replication of a larger (~200 hosts) existing VE (a ”virtual PE”) topology into another VE.

While the replication of a virtual PE into another VE omits the process step of acquiring disk images manually, it enables the automatic generation of test iterations to evaluate consistency between acquired and replicated (and modified) snapshots.

ES 1) The PE replicated in our first evaluation scenario is the Laboratory for IT-Security Architectures (LISA [34]) that also has been the basis for other SME related research [35]. This is a modular, physical testbed explicitly designed for security testing purposes. The configuration of the LISA testbed is well-documented in machine-readable form. This enables a semi-automatic verification of a corresponding VE created via the IRP. The complete interconnected IT-asset topology can be derived from managed network components automatically and host images have been acquired semi-automatically. The only modification in the resulting topology snapshot is the addition of a reference host to conduct tests. The goal of this ES is to create and verify an identical replication in the VE.

ES 2) The VE replicated in our second evaluation scenario was created via the Topology Editor using over 80 endpoints connected via 7 managed network components. The VE configuration is based on the manually composed documentation of a production network and was created in cooperation with a SME. In order to with comply with security policies, pseudonymized layer 3 subnet addresses were deployed in the VE and routing and forwarding policies were adapted analogously. Configuration errors and network inconsistencies (e.g., error-disabled switch ports caused by layer 2 broadcast loops or subnets that could not be reached on layer 3 due to firewall rules or missing routing entries) were introduced manually with the help of the TE when creating modified versions of the original snapshot. The goal of this ES is to create and verify modified snapshots in the VE.

In every replicated snapshot (operated in the VE), a predefined XCCDF reference checklist containing security tests based on the NIST Guideline on Network Security Testing was executed by an additional dedicated *reference host*. In the PE (ES 1), all 55 security tests in the reference checklists resulted in a valid state. Preliminary tests showed that 51 of these tests produced valid result states in the VE. All discrepancies that could be identified were *conceptual discrepancies* regarding performance issues (e.g. iperf throughput-test results and fping[4] round-trip-test results). A corresponding *discrepancy list* to omit tests that cannot be successfully conducted was created and utilized during evaluation of the IRP.

Remaining security tests include, for example, extensive connectivity tests on layer 2, 3 and 4 using different setups of *reference hosts*, vulnerability scans, remote password integrity tests and remote execution of local root kit detection suites. As expected, the initial manual acquisition of block-device images and the preparation of the discrepancy list takes up a significant amount of manual interaction in the IRP. Approx. 80% of the time investment in the first IRP iteration was dedicated to this tasks.

Once the first snapshot has been acquired, creating modified test scenarios for the purpose of security testing (such as adding instances of the included *reference host* with corresponding roles and definitions for OMF experiments) and automatic testing via XCCDF checklists can be configured and initiated solely via the Topology Editor. For this task, no manual input outside the Topology Editor is necessary. In 29 of 30 iterations of the IRP, the Topology Editor proves to be an effective measure to reduce time and costs involved in setting up security tests (excluding the time-intensive initial acquisition procedure).

In ES 1, the initially acquired snapshot was used for every subsequent replication in the VE. The 51 security tests that can be conducted in the VE were used to compare and grade the functionality of modified snapshots after introducing changes. In ES 2, the modifications to snapshots can have severe impacts on the corresponding result states of security tests, e.g. additional packet filter policies on managed network components can result in a failed state of security tests requiring remote connectivity to hosts. As an analogous example, the design and introduction of an appropriate management network (separated via 802.1q tagging) was successfully tested in the VE. Changes to the configuration that represent design changes were then successfully extracted from the VE and operationalized (in maintenance time frames) without a negative impact on functionality or security testing result states in the PE. The evaluation of the IRP shows that typical tests and modifications can be conducted in the VE and tested configuration changes can be deployed in the PE without negative impact. The only impact of the IRP on the PE during the creation of the VE is downtime of indiviual hosts in cases where block-device images have to be extracted manually.

VII. Conclusion & Future Work

The IRP is able to mitigate the typical challenges regarding security testing in SMEs: The IO tool-set enables the acquisition of up-to-date and consistent snapshots representing complete network topologies. The formal representation stored in the IO enables users to incorporate documentation about the network into security test planning. Automation is a core concept of the IRP that is further supported by the use of common security automation protocols and formats, such as SCAP, OVAL or XCCDF. Support of open standards also eases the integration of the IRP into existing automated processes. Considering the trend of operating system manufacturers to include mechanisms able to create block-device snapshots during runtime, the complete IRP can be automated in future work. The manual effort required to acquire disk images in the IRP is still high today and can be mitigated only partially by automated procedures.

We have shown that our approach is sound regarding security testing in SMEs. The inclusion of OVAL and OMF in order to manage security testing procedures also provides access to a variety of community-based test tools and test definitions. In future work, the development of semi-automatic deployment will be addressed to increase the benefits of the IRP for SMEs even further.

Acknowledgment

The work presented here was developed in the context of the project VISA (ID 16BY1160) being funded by the German Federal Ministry of Education and Research.

References

[1] B. Arkin, S. Stender, and G. McGraw, “Software penetration testing,” *Security Privacy, IEEE*, vol. 3, no. 1, 2005.

[4] http://fping.sourceforge.net/

[2] H. Birkholz, I. Sieverdingbeck, C. Bormann, and K. Sohr, "IO: An interconnected asset ontology in support of information security applications," in *7th International Conference, Availability, Reliability and Security*, Prague, Czech, 2012.

[3] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaug, *Technical Guide to Information Security Testing and Assessment*. U.S. DoC, NIST, Gaithersburg, MD, 2008.

[4] M. Rost and K. Bock, "Privacy by design and the new protection goals," 2011.

[5] C. Johnson, *The technical specification for the Security Content Automation Protocol (SCAP)*. U.S. DoC, NIST, Gaithersburg, MD, 2009.

[6] "MITRE." [Online]. Available: http://mitre.org/

[7] S. Quinn, P. Mell, and K. Kent, *The Security Content Automation Program (SCAP): Automating Compliance Checking, Vulnerability Management, and Security Measurement*. U.S. DoC, NIST, Gaithersburg, MD, 2006.

[8] "Oval repository." [Online]. Available: http://oval.mitre.org/repository/

[9] D. Waltermire, C. Schmidt, K. Scarfone, and N. Ziring, *Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2*. U.S. DoC, NIST, Gaithersburg, MD, 2011.

[10] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "OMF: A Control and Management Framework for Networking Testbeds," *SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, 2010.

[11] N. Leavitt, "Is Cloud Computing Really Ready for Prime Time?" *Computer*, vol. 42, no. 1, 2009.

[12] G. Wang and T. S. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *INFOCOM, 2010 Proceedings IEEE*, 2010.

[13] "Open vSwitch project website." [Online]. Available: http://openvswitch.org/

[14] "Open vSwitch feature list." [Online]. Available: http://openvswitch.org/features/

[15] W3C, "OWL 2 Web Ontology Language Document Overview," Tech. Rep., 2009.

[16] D. Beckett and B. McBride, "RDF/XML syntax specification (revised)," *W3C recommendation*, vol. 10, 2004. [Online]. Available: http://www.w3.org/TR/rdf-syntax-grammar/

[17] A. Arnes, P. Haas, G. Vigna, and R. Kemmerer, "Digital forensic reconstruction and the virtual security testbed vise," in *Detection of Intrusions and Malware and Vulnerability Assessment*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4064, pp. 144–163.

[18] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab, "Experience with deter: a testbed for security research," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, 2006, pp. 10 pp.–388.

[19] J. W. Haines, S. A. Goulet, R. S. Durst, and T. G. Champion, "LLSIM: Network Simulation for Correlation and Response Testing," in *IEEE Workshop on Information Assurance*, 2003.

[20] C. Pak and J. Cannady, "Asset priority risk assessment using hidden markov models," in *Proc. of the SIGITE 2009*. New York, NY, USA: ACM, 2009, pp. 65–73.

[21] H. Thompson, "Application penetration testing," *Security Privacy, IEEE*, vol. 3, no. 1, 2005.

[22] L. E. Sanchez, A. S.-O. Parra, D. G. Rosado, and M. Piattini, "Managing Security and its Maturity in Small and Medium-sized Enterprises," *JUCS*, vol. 15, no. 15, 2009.

[23] "IO development website." [Online]. Available: http://blinded-url.invalid/io

[24] M. D. Aime and F. Guasconi, "Enhanced vulnerability ontology for information risk assessment and dependability management," *DEPEND*, pp. 92–97, 2010.

[25] "Zenmap Topology view." [Online]. Available: http://nmap.org/book/zenmap-topology.html

[26] W. Barth, *Nagios: System And Network Monitoring*, ser. No Starch Press Series. Open Source Press, 2006.

[27] "HP Network Management Center." [Online]. Available: http://www8.hp.com/us/en/software-solutions/software.html?compURI=1171412

[28] Gucer, Vasfi and Abbosh, Vincent and Brumfield, Sara C and Marino, Martin and Ross, David and Shah, Ghufran and Turner, Roger, "Deployment Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1," IBM, IBM Redbooks, 2008.

[29] "Cim network model white paper," Distributed Management Task Force, Inc. (DMTF), Tech. Rep., 2003.

[30] "AllegroGraph." [Online]. Available: http://www.franz.com/agraph/allegrograph/

[31] K.-O. Detken, M. Jahnke, H. Birkholz, and C. Dwertmann, "Design and implementation of Virtual Security Appliances (VSA) for SME," *IDAACS 2013*, 2013.

[32] N. Samson, G. Daneels, B. Braem, and C. Blondia, "Overhead analysis of embedded wireless testbeds," in *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2012.

[33] P. Järvinen, "Research Questions Guiding Selection of an Appropriate Research Method," in *European Conference on Information Systems*, Hansen, Bichler, and Mahrer, Eds., Vienna University of Economics and Business Administration, 2000, pp. 124–131.

[34] "LISA-Labor (german)." [Online]. Available: http://www.lisa.fh-dortmund.de

[35] E. Eren and K.-O. Detken, "Identity and Access Management According to the Implementation of the SIMOIT Project and TNC@FHH," *International Journal of Computing*, 2010.