Attack Surface Definitions: A Systematic Literature Review

Christopher Theisen[a], Nuthan Munaiah[b], Mahran Al-Zyoud[c], Jeffrey C. Carver[c], Andrew Meneely[b], Laurie Williams[a]

[a] *North Carolina State University*
[b] *Rochester Institute of Technology*
[c] *University of Alabama*

Abstract

Context: Michael Howard conceptualized the *attack surface* of a software system as a metaphor for risk assessment during the development and maintenance of software. While the phrase *attack surface* is used in a variety of contexts in cybersecurity, professionals have different conceptions of what the phrase means.

Objective: The goal of this systematic literature review is to aid researchers and practitioners in reasoning about security in terms of attack surface by exploring various definitions of the phrase *attack surface*.

Method: We reviewed 644 works from prior literature, including research papers, magazine articles, and technical reports, that use the phrase *attack surface* and categorized them into those that provided their own definition; cited another definition; or expected the reader to intuitively understand the phrase.

Results: In our study, 71% of the papers used the phrase without defining it or citing another paper. Additionally, we found six themes of definitions for the phrase *attack surface*.

Conclusion: Based on our analysis, we recommend practitioners choose a definition of *attack surface* appropriate for their domain based on the six themes we identified in our study.

Keywords: attack surface, vulnerabilities, software engineering, systematic literature review

1. Introduction

Vulnerabilities in software are an unfortunate, but inevitable, reality. As a result, software development organizations must take a proactive approach to security. The Trustworthy Computing Security Development Lifecycle (SDL) introduced at Microsoft in 2004 [1] is one such instance of a proactive approach to software security. One of the elements in the design phase of SDL was the measurement of the *attack surface* of software. At the time, the phrase software attack surface was fairly unfamiliar, having been introduced only a year earlier by Michael Howard [2].

Software security researchers and professionals have used Howard's concept of the attack surface to discuss the overall security posture of a system, or the effectiveness of a given security measure. For example, reducing the attack surface is one way researchers provide evidence that the system is more secure. Practitioners can also use attack surface measurements to prioritize their fortification efforts.

But what is an *attack surface*, exactly? A variety of definitions exist for the phrase, which drives how researchers conduct their measurements. The varying definitions result in confusion when professionals and researchers have different views on what the phrase *attack surface* means. Having a consistent language with which to talk about the attack surface of software systems would help focus discussions on how to address each different definition of attack surface. A sampling of the existing definitions define the attack surface as follows:

- "...union of code, interfaces, services, protocols, and practices available to all users, with a strong focus on what is accessible to unauthenticated users." [3]
- "...the system's actions that are externally visible to its users and the system's resources that each action increases or modifies." [4]
- "...a list of attack features: Open sockets, Open RPC endpoints, Open named pipes, Services, etc." [2]

Having a seminal definition (or a set of definitions) to reference as the "official" attack surface definition(s) would help clarify discussions about and measurements of attack surfaces.

The goal of this systematic literature review is to aid researchers and practitioners in reasoning about security in terms of attack surface by exploring various definitions of the phrase *attack surface*.

The research questions we address in this work are:

RQ1 Diversity How is the phrase *attack surface* used by researchers?

RQ2 Variety What are the different definitions of the phrase *attack surface* in the research literature, and how frequently is each used?

RQ3 Unification Based upon themes of attack surface definitions, can a unified definition of the phrase *attack surface* be determined?

To achieve this goal, we performed a systematic literature review of the use of the phrase *attack surface* in literature, including research papers, magazine articles, and technical reports. These writings represent the opinion of a variety of professionals in both an academic and industrial context, typically in the computer security domain. After identifying a set of 1,433 potential papers for inclusion in our study, we selected 644 that used the phrase *attack surface* in some way in the text. For these papers, we determined whether they provided their own definition, cited another definition, or expected the reader to understand their use of the phrase intuitively. We identified the source of each paper, such as a specific database or aggregator. Additionally, we analyzed the level of granularity of each paper's use of attack surface. For example, some papers discussed the attack surface of an entire network of systems, while others discussed how specific functions affect the attack surface of one system. We noticed six themes based on concept similarity and recommend a seminal definition based on frequency of citations, citations in important works, and the expert judgment of the attack surface researchers involved with this review.

Recent work using the attack surface concept varies, with some prioritizing

the localization of vulnerabilities, others identifying vulnerable code, and others determining how risky specific vulnerabilities are. Younis et al. [5] analyzed the relationship between the attack surface of Apache HTTP Server and the density of vulnerabilities in the system. Munaiah et al. [6] used call graphs to determine the proximity of security vulnerabilities to the attack surface of the software system and found that vulnerabilities were found near the surface of the target system. Theisen et al. [7] developed Risk-Based Attack Surface Approximation (RASA), which uses crash dump stack traces to estimate the attack surface of a target system.

The rest of this paper is organized as follows: Section 2 presents work related to our study. Section 3 describes the methodology used to perform this review. Section 4 describes the results of our study. Section 5 discusses our results and provides a set of conclusions about our study.

2. Related Work

The methodology for conducting this SLR is based on previous SLRs and guidelines for performing SLRs from other researchers in the field. Zhang et al. [8] suggested the use of a Quasi-gold Standard set of papers to validate search terms when searching for papers to include in a literature review. Kitchenham et al. [9] analyzed SLRs in Software Engineering, eventually leading to a list of recommendations for good SLR practices in 2013 [10]. They suggest setting concrete inclusion and exclusion criteria for papers in the study corpus, among other suggestions. Kitchenham also remarks on the lack of tools to support SLRs. As part of this study, we make available the scripts used in support of generating the corpus found in this study with the hope that it helps future authors with their own work.

The phrase *attack surface* was used before it became popular for describing a measure of security for software systems. In 1972, Cortes filed a patent describing a drilling tool that used the phrase *attack surface* [11]. In the patent, *attack surface* was used to describe the tip of the drill, and describes how that tip might

be reinforced with diamonds for durability and increased cutting power. In 2003, Michael Howard introduced the phrase *attack surface* in an MSDN Magazine Article [2], which led to further research in the area by Howard, Manadhata, and Wing [4, 12, 13]. Current work in the area of *attack surface* focuses on creating empirical and theoretical measures for the *attack surface* of a software system or computer network [7, 14, 6, 5, 15].

One of the practical applications of the phrase *attack surface* is its use in the Common Weakness Scoring System (CWSS). CWSS uses the attack surface to group a set of metrics together for use in the CWSS score. These metrics are:

- *Required Privilege (RP)* - The type of privileges that an attacker must already have in order to reach the code/functionality that contains the weakness.
- *Required Privilege Layer (RL)* - The operational layer to which the attacker must have privileges in order to attempt to attack the weakness.
- *Access Vector (AV)* - The channel through which an attacker must communicate to reach the code or functionality that contains the weakness.
- *Authentication Strength (AS)* - The strength of the authentication routine that protects the code/functionality that contains the weakness.
- *Level of Interaction (IN)* - The actions that are required by the human victim(s) to enable a successful attack to take place.
- *Deployment Scope (SC)* - Whether the weakness is present in all deployable instances of the software, or if it is limited to a subset of platforms and/or configurations.

Each of these metrics is assigned a weight and aggregated to provide an overall attack surface scoring for the weakness. The CWSS Attack Surface score has several parallels with the definitions of attack surface provided in this document.

3. Methodology

In the subsections that follow, we describe each step in the approach to identifying the papers that were relevant to the research goal of our systematic literature review. At a high-level, our approach comprised of the following steps, organized into four stages:

Selection

- Enumerate source(s) of studies
- Identify search keyword(s)
- Collect studies using the search keyword(s) from the source(s) enumerated

Inclusion and Exclusion

- Identify relevant papers using inclusion and exclusion criteria

Categorization

- Categorize the relevant papers based on usage of the phrase *attack surface*

Theme Identification

- Identify themes of definitions of the phrase *attack surface* based on conceptual similarity

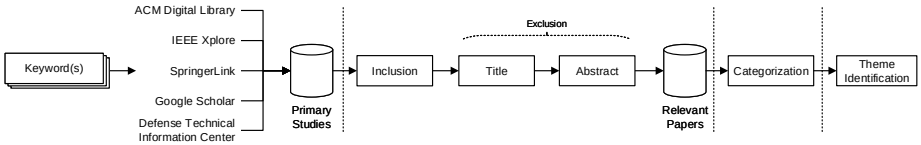

Figure 1: Pictorial overview of the flow through various stages in the SLR

Our methodology is inspired by the guidelines for performing SLRs prescribed by Kitchenham and Brereton [10].

Shown in Figure 1 is a pictorial overview of the various stages in our SLR.

3.1. Selection

In this stage, the various sources of studies relevant to our SLR and the keywords used to search for these studies are identified.

3.1.1. Enumerate source(s) of studies

The first step in our review was to enumerate the primary study sources. The digital libraries are the most commonly used sources of academic content in SLRs. However, to achieve a higher coverage of a particular niche area of research, papers from specific journals or conference proceedings and/or technical reports published by universities or a sponsoring agency have to be included.

In our study, we considered the following sources of academic content:

1. ACM Digital Library[1]
2. IEEE Xplore Digital Library[2]
3. SpringerLink[3]
4. Google Scholar[4]
5. Defense Technical Information Center (DTIC[5])

We considered ACM Digital Library and IEEE Xplore Digital Library because they are the primary publishers of proceedings of conferences that cater to security metric research. We considered SpringerLink because it captured several journals that publish articles related to attack surface literature. Although Google Scholar is not a *source* of academic content, we have included it in our list because Google Scholar indexes a wide variety of academic and technical content (such as white papers published by organizations) that may be relevant to our study. We include DTIC because several papers included in our quasi-gold standard set were found through the DTIC engine.

[1] `http://dl.acm.org/`
[2] `http://ieeexplore.ieee.org/`
[3] `http://link.springer.com/`
[4] `http://scholar.google.com`
[5] `http://www.dtic.mil/`

3.1.2. Identify search keyword(s)

The source(s) of studies identified in the previous step support advanced searching of their databases. The next step in the selection stage is to identify keywords that may be used to search for studies. We used an iterative approach to propose a candidate search keyword and validate its ability to retrieve studies. We used the concept of Quasi-gold Standard proposed by Zhang et al. [8] to validate and refine our list of candidate search keywords. The principle in quasi-gold standard is to compose a set (called the quasi-gold standard set) of studies that are known to be relevant to the SLR. The candidate search keywords are used to retrieve studies in the quasi-gold standard set. The iteration terminates when all the studies in the quasi-gold standard set can be retrieved using the search keywords identified.

Since two of the authors of this paper are software security researchers working almost exclusively on research related to the concept of attack surfaces, our quasi-gold standard set was composed of the attack surface literature cited by prior publications [7, 6] of the two authors. The list of the 17 studies in our quasi-gold standard set is presented in Appendix A.

The phrase *attack surface* is fairly generic; for instance, we found a patent using the phrase *attack surface* in describing a drilling tool [11]. To constrain the scope of the search, we logically appended the fixed keyword—`security`—to a search keyword using the `AND` operator. In the case of Google Scholar, we had to introduce an additional fixed keyword—`entry points`—to constrain the number of search results, as accessing the 99^{th} page of search results caused a server error.

The logical combination of the search keywords using the `AND` operator results in a search string. We iterated through our search strings four times and arrived at a set of two different search strings. In summary, we used the string `"security" AND "attack surface"` when searching ACM Digital Library, Defense Technical Information Center (DTIC), IEEE Xplore Digital Library, and SpringerLink and the string `"security" AND "attack surface" AND "entry`

`points"` when searching Google Scholar.

3.1.3. Collect studies using the search keyword(s) from the source(s) enumerated

The next step in the selection stage is to use the search keyword(s) identified in the previous step and search for studies in each of the source(s) identified in the first step. As highlighted in prior literature [10], tools to support large-scale SLRs are rare. As a supplementary contribution of our work, we have developed an open-source utility, called `SLRUtility`, to support the collection of studies from various academic sources. The source code for `SLRUtility` is available on GitHub at `https://github.com/theisencr/SLRUtility`.

The last run of our collection step was on June 1, 2016 for Google Scholar and May 31, 2016 for the other four sources. All studies collected from each of the sources are combined, automatically detecting and eliminating title-based duplicates (if any) across the different sources.

3.2. Inclusion and Exclusion

In this stage, the studies collected in the previous stage are subject to an inclusion and an exclusion criteria to filter out those studies that may not be relevant to our SLR. This stage is critical to ensure that a sizable, yet manageable, number of studies are selected. The studies that pass this stage will be considered relevant to our SLR and will be referred to as *relevant papers* in the remainder of the paper. The relevant papers are the ones from which we obtain the data needed to address the research questions in our SLR.

3.2.1. Identify relevant papers using inclusion and exclusion criteria

We identified inclusion and exclusion criteria for whether or not papers are pertinent to our study. The criteria used in our study are presented below. These criteria were applied before and during each categorization and filtering step in the following sections.

Inclusion Criteria

Include a study if,

- The study uses the phrase *attack surface*.
- The study is written in English.
- The study is in the software security domain.
- The study is full length research paper (i.e. not a presentation or a supplement to a poster).
- The study was published in or after the year 2000. While attack surface was formally defined by Howard in 2003, other researchers have used the phrase informally since 2000.

Exclusion Criteria

Exclude a study if,

- *Title*: The title provides sufficient evidence to indicate that the study is not related to cybersecurity. For example, a study titled *Unmaking the Dark Continent: South Africa, Africa and the Image Make-Over Narrative in the South African Press* was one of the papers decided to be excluded based on the title.
- *Abstract*: The abstract provides sufficient evidence to indicate that the study is not related to cybersecurity. For example, a study titled *Energy Theft in the Advanced Metering Infrastructure* was one of the papers decided to be excluded based on the abstract of the paper.

In the inclusion criteria, we chose to limit the scope of search to literature published after the year 2000. We chose the year 2000 based on a steep increase in the use of the phrase *attack surface* as observed in the plot obtained from Google Ngram Viewer.[6] Additionally, SLRs tend to impose a restriction on the number of pages that a study must have to be included. We, however, impose no such restriction as the goal of our review is to understand the usage of the phrase *attack surface* and the length of the study may not be a relevant factor.

[6] `https://books.google.com/ngrams`

The inclusion criteria is fairly straightforward and can largely be automated. Furthermore, any false positives (i.e. studies that are irrelevant but considered relevant) that may creep through the inclusion criteria are likely to be found and excluded during the application of the exclusion criteria.

We note that, in applying the inclusion criteria, we realized that all peer-reviewed papers indexed by DTIC were also indexed by SpringerLink. As a result, we chose to combine the duplicates from DTIC and SpringerLink in subsequent stages of our methodology. The removal of duplicates left 959 papers for review.

The exclusion criteria, being manual, is inherently subjective. We mitigated the potential for bias due to the subjectivity by having at least two authors independently apply the exclusion criteria to the same set of studies. We further used the inter-rater reliability measure—Cohen's κ [16]—to quantify the level of agreement between the two authors who applied the exclusion criteria. Cohen's κ quantifies the level of agreement as it accounts for agreements that occur by chance. The disagreements, if any, were resolved by the authors presenting their case for excluding a paper. In cases where consensus could not be reached, we decided to include the paper since, in the next stage (i.e. categorization), a full text reading would provide more information on the relevance of the study to our SLR.

The set of studies that pass through the inclusion and exclusion criteria are considered relevant to our SLR.

3.3. Categorization

In this stage, the full text of the relevant papers identified in the previous stage are read and categorized into categories that are based on papers' usage context of the phrase *attack surface*. Since the goal of our study was to understand the various definitions of the phrase *attack surface* used in the community, we categorized the relevant papers into one of four categories. Each paper fits into exactly one category.

- `Define` - The paper defines, formally or informally, the phrase *attack surface* in the text. Papers in prior attack surface literature or papers that include a glossary with a definition of the phrase *attack surface* tend to be placed into this category.

- `Supported Use` - The paper uses the phrase *attack surface* without explicitly defining it. However, the usage of the phrase is supported by citing a paper that defines the phrase.

- `Unsupported Use` - The paper uses the phrase *attack surface* with neither an explicit definition for it nor a citation to a paper that defines the phrase.

- `Not Relevant` - The paper is not relevant to our SLR. Papers that are categorized into the `Not Relevant` category are false positives from the previous stage.

The order of the categories presented above is the order in which the categorizations were made. For instance, if a paper is extending an existing (cited) definition of the phrase *attack surface*, the paper will be categorized as `Define` and not `Supported Use`.

When reading the full text of relevant paper for categorized, we also captured the association (through citation) between papers categorized as `Supported Use` and those categorized as `Define`. We also captured self-associations between papers categorized as `Define` in cases where a paper was extending or modifying an existing (cited) definition.

In addition to categorizing the papers based on the usage of the phrase *attack surface*, we categorized papers in the `Define` and `Supported Use` categories based on the granularity at which the phrase was used. We propose the following categories for the papers based on the granularity of phrase usage:

- `Function` - The paper uses the phrase *attack surface* with methods, functions or individual lines of code as the lowest unit of reasoning. For example, a paper that discusses the attack surface implications of allowing

certain set of functions to be accessible through the Application Program Interface (API).

- `File` - The paper uses the phrase *attack surface* as applied to source code files. For example, a paper that discusses the attack surface implications of having source code in certain files vulnerable to particular type of attack.

- `Binary` - The paper uses the phrase *attack surface* with source code packages such as binaries, packages, modules or components. For example, a paper that presents different approaches to reduce the attack surface of a binary.

- `System` - The paper uses the phrase *attack surface* when reasoning about entire systems. For example, a paper that presents the security implications of enabling certain features in an operating system such as Windows and Linux.

- `Computer Network` - The paper uses the phrase *attack surface* as applied to entire networks. For example, a paper evaluating the notion of isolating certain set of sensitive hosts to a sub-network, inaccessible, in general, to hosts outside of the network.

- `Theoretical` - The paper uses the phrase *attack surface* in a theoretical capacity. Papers that are categorized into this level of granularity typically attempt to quantify the attack surface of an entity based on theoretical notions.

  We also used `Theoretical` to categorize papers when we could not infer the granularity from the full text.

As with the previous stage, the categorization of relevant papers into the categories of attack surface usage and the granularity of the phrase usage was primarily a manual approach. Here again, we had two authors independently categorize the same set of papers into different categories while using Cohen's κ to quantify the level of agreement. We did not use Cohen's κ for assessing

agreements with associations captured because of the limited potential for subjectivity in identifying citation information from either the references list or the footnote.

3.4. Theme Identification

The final stage in our SLR was to analyze and identify themes in various definitions of the phrase *attack surface* from the relevant papers categorized as `Define`. We only considered those papers that has at least one citation at the time of our study. Considering only papers that had at least one citation resulted in 19 papers to identify themes.

As with other manual steps, two authors independently analyzed each definition and assigned it to an appropriate theme. Identifying a theme of a definition indicates the conceptual similarity among one or more definitions. The authors generated their own understanding of relationships between different definitions and created a set of themes they saw in the set of cited definitions. After generating their own themes, the authors involved met to discuss their respective conclusions. The discussion concluded with both authors agreeing on six themes and an name for each theme. The final set of themes were reviewed by a third author for validation.

4. Results

In the subsections that follow, we address the research questions in our SLR using the data collected by applying our methodology. We first present the details of applying different stages of the methodology to compose our data set of relevant papers from the studies obtained from the different source(s). The pictorial overview of the flow through the various steps shown in Figure 2 has annotations showing the number of studies at the end of each stage of the methodology. The figure also shows the Cohen's κ quantifying the level of agreement between the authors in applying the exclusion criteria to studies and in the categorization of the relevant papers.

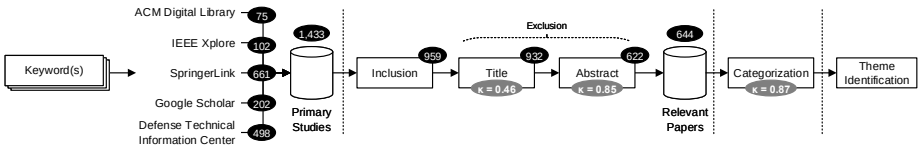

Figure 2: Number of studies at the end of the selection stage and at the end of each step of the inclusion and exclusion stage

4.1. Title Filtering

After reviewing the titles of all 959 papers individually, the authors met to resolve disagreements over categorization. The authors disagreed on 47 of the 959 papers for categorization. After resolving these differences, the authors agreed on the removal of 27 of the 959 papers from literature review, leaving 932 papers for the next step of the filtering process. Examples of titles that were removed from consideration include "Correlation is not Causation", "Clearance of Flight Control Laws for Carefree Handling of Advanced Fighter Aircraft," and "Plan X and Generation Z."

The inter-rater reliability between the two authors in the title filtering stage was $\kappa = 0.46$.

4.2. Abstract Filtering

The authors reviewed the abstracts of the remaining 932 papers individually, categorizing each for inclusion or exclusion from our corpus. The authors then met to resolve disagreements over the inclusion or exclusion of each paper. The authors agreed that 622 papers should be included in the final corpus for the study. Four of the papers were determined to be duplicated in our corpus or did not have full text available on their respective database, and were not categorized. Over the course of the categorization of papers, 22 additional papers were added, as they were cited by other papers for their definition of attack surface, bringing the final total of papers to categorize to 644.

The inter-rater reliability between the two authors in the abstract filtering stage was $\kappa = 0.85$.

4.3. RQ1: Diversity

Question: How is the phrase attack surface used by researchers?

To answer this question, we look at the two categorizations made for each paper in our final, filtered corpus; the categorization of the use of the phrase *attack surface*, and the level of granularity at which the phrase was discussed.

Shown in Figure 3 is the distribution of papers by their use of the phrase *attack surface*.

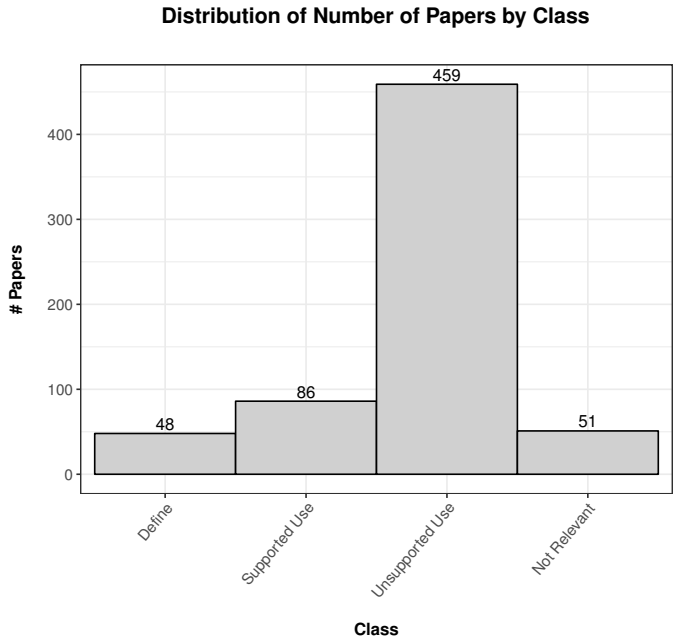

Figure 3: Distribution of number of papers by class

In our results, 459 of the 644 papers (or 71%) in the final corpus were categorized as `Unsupported Use`, or provided no external support of their use of attack surface in the text of the document. Authors not providing support for their use of the phrase indicates a lack of understanding of the different possible definitions, or an assumption that the audience of their paper have the same definition in mind. From the same data, we see there are 48 papers categorized as `Define`, or providing a definition of attack surface. These 48 papers provide

48 definitions of attack surface in the literature with varying degrees of overlap and authority. Many of these definitions are not themselves cited. Therefore, we conclude that we cannot assume a canonical definition of attack surface exists that the community has agreed on, as different individuals have varying opinions on what the definition is.

By contrast, 86 papers (or 13%) in our corpus provided `Supported Use`, or a citation or a footnote of their use of the phrase *attack surface*. These papers skewed towards security focused works. We observed that the more closely related the research was to the topic of attack surfaces, the more likely it was that the researcher supported their definition of attack surface in some way.

Shown in Figure 4 is the distribution of papers by the level of granularity at which the phrase *attack surface* is used.

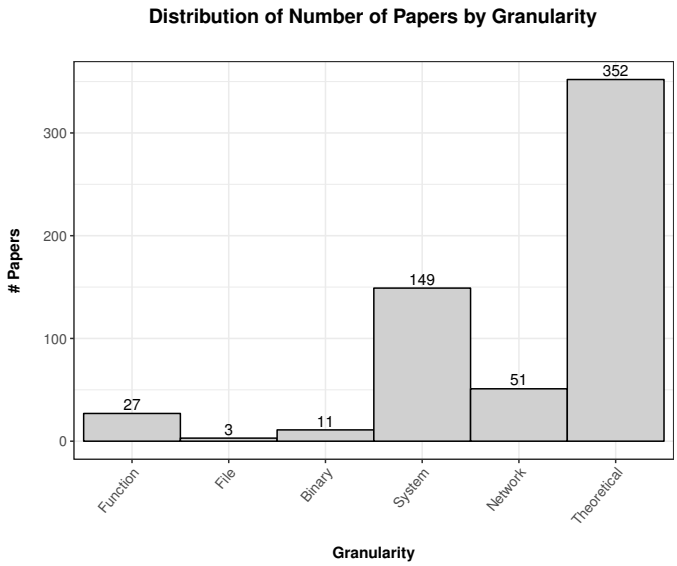

Figure 4: Distribution of number of papers by granularity

In our corpus, 352 of the 644 (or 55%) papers discussed attack surfaces at the `Theoretical` level, while 149 of the papers discussed the phrase at a Software `System` level. Only 41 papers discussed attack surfaces at a `Binary`, `File`, or `Function` level, possibly indicating that software development work represented

a minority of the papers in our corpus. The high percentage of theoretical uses of the phrase indicates that attack surfaces are still mostly considered a theoretical exercise, with limited number of researchers or practitioners in our corpus applying the concept to a real software system or computer network.

> The phrase *attack surface* is used without an associated definition in 71% of papers in our corpus. Attack surface is used as a theoretical concept in 55% of the papers in our corpus, rather than in reference to specific source code.

4.4. RQ2: Variety

Question: What are the different definitions of the phrase attack surface *in the research literature, and how frequently are they used?*

In this research question, we enumerate the different definitions of the phrase *attack surface* as presented in papers from prior literature that almost exclusively reason about computer security in terms of attack surfaces.

We found a total of 48 papers authored by 33 different first authors that include a (formal or informal) definition of the phrase *attack surface*. Overall, 19 papers were cited in support of the phrase *attack surface*. The most frequently cited work was *An Attack Surface Metric* by Manadhata [17] with 43 unique citations. However, this paper cites several other papers in support of its own definition of attack surface, primarily the *Measuring Relative Attack Surfaces* work by Howard et al. [12] The complete list of papers that define attack surface is presented in Appendix B.

> We identified 48 different definitions of the phrase *attack surface*, with the most frequently cited definition cited 43 times within our set of attack surface papers.

4.5. RQ3: Unification

Question: Based upon themes of attack surface definitions, can a unified definition of the phrase attack surface *be determined?*

As described in Section 3.4, we group papers from Appendix B that were cited at least once for their definition of the phrase *attack surface* into themes based on conceptual similarity. From Appendix B, 19 papers were cited at least once. We noticed six themes representing interpretations of the phrase *attack surface*. The six themes are described below. The full definition of attack surface from each paper and the associated themes can be found in Table C.1 in Appendix C.

- *Methods:* The attack surface is the methods of implementation, data channels, and data present in the system, with no specific attack features mentioned.

The *Methods* theme is a programming-centric definition that focuses on program flow through a software system. Such an attack surface could be measured by programs like GNU cflow or other static analysis tools for measuring connections through software systems.

- *Adversaries:* The attack surface is the union of all possible ways an attacker could cause damage to a system.

The *Adversaries* theme focuses on attacker behavior only, as only points in the system with active attacks would be considered part of the attack surface. As an example to differentiate the *Methods* theme from the *Adversaries* theme, there could be entry points from the *Methods* theme that have no equivalent in the *Adversaries* theme due to a lack of feasible attacks.

- *Flows:* The attack surface is defined as data flow and control flow only, without considering methods or avenues of attacks.

The *Flows* theme is based on user behavior. The *Flows* theme would be a subset of the *Methods* theme, in that some paths available in the *Methods* theme may be unavailable to a subset of users. The *Flows* theme attack surface would change as user authorization level changes.

- *Features:* The attack surface is a enumeration of all available attack avenues to a target system.

The *Features* theme is a higher level set of definitions that focus on the features available in a system at the level of functionality. This differs specifically from the previous definitions in that it encapsulates functionality in a system rather than the paths data takes through a system. This theme requires a specific enumeration of parts of the system that are possible attack features, such as a list of open ports, a list of services running by default, et cetera. This is distinct from the methods theme, as specifics are left ambiguous in the methods theme. As an example, a Windows machine with 50 running services would have a higher attack surface than a Windows machine with 30 running services.

- *Barriers:* The attack surface is the method of preventing attacks, rather than the paths attacks can occur on, by malicious parties.

The *Barriers* theme focuses on preventative efforts in security on a system, such as firewalls or security policies. This is independent from the other themes as it focuses on defensive behavior, rather than possible points of attack.

- *Reachable Vulnerabilities:* The attack surface is the vulnerabilities that are exposed to end users via paths or flows, rather than the paths or flows themselves.

Finally, the *Reachable Vulnerabilities* theme is focused on the exposure of vulnerabilities that attackers can exploit in a software system. Using this theme, a system with no vulnerabilities does not have an attack surface.

Based on our results, we recommend that researchers and practitioners choose a definition of attack surface that most closely matches the domain that they are using the phrase in. While we consider Howard et al. definition from *Measuring Relative Attack Surfaces* [12] in the `Methods` theme the canonical definition of the phrase *attack surface*, this definition may not be appropriate

in all cases. We base this conclusion on wide variety of the use of the phrase *attack surface* based on our results for RQ1 and RQ2, along with the six different themes that resulted from our analysis. By using more specific definitions of *attack surface*, practitioners and researchers can speak more precisely about the phrase. We recommend that practitioners use domain specific language, such as talking directly about server architecture or software architecture, when defining the context in which they are using the phrase *attack surface*.

When talking about attack surfaces in a theoretical context, the Howard definition provides a definition encompassing attack surface definitions included in the other five themes of definitions included in our study. In *Measuring Relative Attack Surfaces* by Howard et al. [12], the phrase *attack surface* is defined as being along three dimensions defined in the paper, and replicated directly below:

- *Targets and Enablers:* To achieve his goal, the adversary has in mind one or more targets on the system to attack. An *attack target*, or simply *target*, is a distinguished process or data resource on *System* that plays a critical role in the adversary's achieving his goal. We use the phrase *enabler* for any accessed process or data resource that is used as part of the means of the attack but is not singled out to be a target.

- *Channels and protocols: Communication channels* are the means by which the adversary gains access to the targets on *System*. We allow both message-passing and shared-memory channels. *Protocols* determine the rules of interaction among the parties communicating on a channel.

- *Access rights.* These rights are associated with each process and data resource of a state machine.

Howard et al. go on to state that as each of these dimensions grow, the attack surface of the target grows as well. Another important aspect that the paper by Howard et al. addresses is the notion of measurement of attack surface, in that, the approach used to measure the attack surface is as important as the

dimensions used to represent the attack surface. The metrics used to measure attack surface can vary. Our other themes are an indication of possible metrics and measurement techniques. The `Adversaries` theme focuses on potential damage as measured by data loss and attacks. The `Flows` theme focuses on how data moves through systems. The `Features` theme provides a list of software features that could be used to measure the attack surface of a system. The `Reachable Vulnerabilities` theme uses vulnerabilities themselves as a metric for measuring the attack surface. These different measurements indicate that a singular definition would likely have deficiencies for one or all of the themes found in our study.

> We recommend researchers and practitioners use one of the six themes of definitions we found as part of this study (Methods, Adversaries, Flows, Features, Barriers, Reachable Vulnerabilities) as their definition of attack surface with proper citations, while taking care to ensure that the context in which their definition applies is explicitly defined, such as networking or software.

5. Discussion

In this SLR, we categorized a total of 644 papers related to the topic of attack surface. We determined the frequency with which the definitions of attack surface used in these papers is based on a citation, and determined the most frequently cited definitions for the phrase *attack surface*. Based on our criteria, we recommend that researchers and practitioners choose an attack surface definition from one of the six identified themes with context-specific clues.

One of the discussion points in this literature review is the determination of when something has become "common knowledge" and no longer needs to be cited. For example, in many cases, the phrase "security vulnerability" is not cited in security related work, yet different definitions for the phrase exist. Along the same vein, can we consider the phrase attack surface "common knowledge?" While we have determined from our review that it is not considered common

knowledge, when does something rise to the level of common knowledge? Will we reach a point where the phrase *attack surface* is common enough to be used without citations in research?

Acknowledgements

This research was sponsored by the U.S. National Security Agency Science of Security Lablet at North Carolina State University under grant H98230-14-C-0139. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

References

[1] S. Lipner, The Trustworthy Computing Security Development Lifecycle, in: 20th Annual Computer Security Applications Conference, 2004, pp. 2–13. doi:10.1109/CSAC.2004.41.

[2] M. Howard, Fending Off Future Attacks by Reducing Attack Surface, MSDN Magazine, 2003.
URL https://msdn.microsoft.com/en-us/library/ms972812.aspx

[3] M. Howard, Attack Surface: Mitigate Security Risks by Minimizing the Code You Expose to Untrusted Users, MSDN Magazine, 2004.
URL http://msdn.microsoft.com/en-us/magazine/cc163882.aspx

[4] P. Manadhata, J. M. Wing, Measuring a System's Attack Surface, Tech. Rep. CMU-CS-04-102, Carnegie Mellon University (Jan 2004).
URL http://reports-archive.adm.cs.cmu.edu/anon/2004/CMU-CS-04-102.pdf

[5] A. A. Younis, Y. K. Malaiya, I. Ray, Using Attack Surface Entry Points and Reachability Analysis to Assess the Risk of Software Vulnerability

Exploitability, in: 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering, 2014, pp. 1–8. doi:10.1109/HASE.2014.10.

[6] N. Munaiah, A. Meneely, Beyond the Attack Surface: Assessing Security Risk with Random Walks on Call Graphs, in: Proceedings of the 2016 ACM Workshop on Software PROtection, SPRO '16, ACM, New York, NY, USA, 2016, pp. 3–14. doi:10.1145/2995306.2995311.
URL http://doi.acm.org/10.1145/2995306.2995311

[7] C. Theisen, K. Herzig, P. Morrison, B. Murphy, L. Williams, Approximating Attack Surfaces with Stack Traces, in: Proceedings of the 37th International Conference on Software Engineering - Volume 2, ICSE '15, IEEE Press, Piscataway, NJ, USA, 2015, pp. 199–208.
URL http://dl.acm.org/citation.cfm?id=2819009.2819039

[8] H. Zhang, M. A. Babar, P. Tell, Identifying relevant studies in software engineering, Information and Software Technology 53 (6) (2011) 625–637. doi:http://dx.doi.org/10.1016/j.infsof.2010.12.010.
URL http://www.sciencedirect.com/science/article/pii/S0950584910002260

[9] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - A systematic literature review, Information and Software Technology 51 (1) (2009) 7–15. doi:10.1016/j.infsof.2008.09.009.
URL http://dx.doi.org/10.1016/j.infsof.2008.09.009

[10] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Information and Software Technology 55 (12) (2013) 2049–2075. doi:10.1016/j.infsof.2013.07.010.
URL http://dx.doi.org/10.1016/j.infsof.2013.07.010

[11] C. A. C., Drilling tool with elements having diamond-studded attack sur-

face, uS Patent 3693735 (Sep 1972).
URL https://www.google.ch/patents/US3693735

[12] M. Howard, J. Pincus, J. M. Wing, Measuring Relative Attack Surfaces, in: Proceedings of Workshop on Advanced Developments in Software and Systems Security, 2003.

[13] M. Howard, D. LeBlanc, Writing Secure Code, Best Practices Series, Microsoft Press, 2003.
URL https://books.google.com/books?id=fzsFCAAAQBAJ

[14] C. Theisen, K. Herzig, B. Murphy, L. Williams, Risk-based attack surface approximation: how much data is enough?, in: Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 2017 IEEE/ACM 39th International Conference on, IEEE, 2017, pp. 273–282.

[15] A. A. Younis, Y. K. Malaiya, Using Software Structure to Predict Vulnerability Exploitation Potential, in: 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion, 2014, pp. 13–18. doi:10.1109/SERE-C.2014.17.

[16] J. Cohen, Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit., Psychological bulletin 70 (4) (1968) 213.

[17] P. K. Manadhata, J. M. Wing, An Attack Surface Metric, IEEE Transactions on Software Engineering 37 (3) (2011) 371–386. doi:10.1109/TSE.2010.60.

[18] P. Manadhata, J. Wing, M. Flynn, M. McQueen, Measuring the Attack Surfaces of Two FTP Daemons, in: Proceedings of the 2Nd ACM Workshop on Quality of Protection, QoP '06, ACM, New York, NY, USA, 2006, pp. 3–10. doi:10.1145/1179494.1179497.
URL http://doi.acm.org/10.1145/1179494.1179497

[19] P. K. Manadhata, Y. Karabulut, J. M. Wing, Report: Measuring the Attack Surfaces of Enterprise Software, Springer Berlin Heidelberg, Berlin,

Heidelberg, 2009, pp. 91–100. doi:10.1007/978-3-642-00199-4_8.
URL http://dx.doi.org/10.1007/978-3-642-00199-4_8

[20] P. K. Manadhata, An Attack Surface Metric, Ph.D. thesis, School of Computer Science (Nov 2008).
URL http://reports-archive.adm.cs.cmu.edu/anon/2008/CMU-CS-08-152.pdf

[21] P. K. Manadhata, K. M. C. Tan, R. A. Maxion, J. M. Wing, An Approach to Measuring A System's Attack Surface, Tech. Rep. CMU-CS-07-146, Carnegie Mellon University (Aug 2007).
URL http://reports-archive.adm.cs.cmu.edu/anon/2007/CMU-CS-07-146.pdf

[22] M. Howard, J. Pincus, J. M. Wing, Measuring Relative Attack Surfaces, Springer US, Boston, MA, 2005, pp. 109–137. doi:10.1007/0-387-24006-3_8.
URL http://dx.doi.org/10.1007/0-387-24006-3_8

[23] A. A. Younis, Y. K. Malaiya, Relationship between Attack Surface and Vulnerability Density: A Case Study on Apache HTTP Server, in: Proceedings on the International Conference on Internet Computing (ICOMP), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 1.

[24] D. Brenneman, Improving Software Security by Identifying and Securing Paths Linking Attack Surface to Attack Target, Tech. rep., McCabe Software Inc. (2012).
URL http://www.mccabe.com/pdf/Identifying%20and%20Securing%20Paths%20Linking%20Attack%20Surfaces%20to%20Attack%20Targets.pdf

[25] A. Bartel, J. Klein, Y. Le Traon, M. Monperrus, Automatically Securing Permission-based Software by Reducing the Attack Surface: An Application to Android, in: Proceedings of the 27th IEEE/ACM International

Conference on Automated Software Engineering, ASE 2012, ACM, New York, NY, USA, 2012, pp. 274–277. doi:10.1145/2351676.2351722.
URL http://doi.acm.org/10.1145/2351676.2351722

[26] L. Wang, S. Jajodia, A. Singhal, S. Noel, k-Zero Day Safety: Measuring the Security Risk of Networks against Unknown Attacks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 573–587. doi:10.1007/978-3-642-15497-3_35.
URL http://dx.doi.org/10.1007/978-3-642-15497-3_35

[27] A. Hahn, M. Govindarasu, Cyber Attack Exposure Evaluation Framework for the Smart Grid, IEEE Transactions on Smart Grid 2 (4) (2011) 835–843. doi:10.1109/TSG.2011.2163829.

[28] P. Parrend, S. Frenot, Security benchmarks of OSGi platforms: toward Hardened OSGi, Software: Practice and Experience 39 (5) (2009) 471–499. doi:10.1002/spe.906.
URL http://dx.doi.org/10.1002/spe.906

[29] P. Finnigan, User Security, Apress, Berkeley, CA, 2010, pp. 467–505. doi:10.1007/978-1-4302-2669-7_14.
URL http://dx.doi.org/10.1007/978-1-4302-2669-7_14

[30] P. K. Manadhata, D. K. Kaynar, J. M. Wing, A Formal Model for A System's Attack Surface, Tech. Rep. CMU-CS-07-144, Carnegie Mellon University (July 2007).
URL http://reports-archive.adm.cs.cmu.edu/anon/2007/CMU-CS-07-144.pdf

[31] Y. Huang, A. K. Ghosh, Introducing Diversity and Uncertainty to Create Moving Attack Surfaces for Web Services, Springer New York, New York, NY, 2011, pp. 131–151. doi:10.1007/978-1-4614-0977-9_8.
URL http://dx.doi.org/10.1007/978-1-4614-0977-9_8

[32] J. Bickford, H. A. Lagar-Cavilla, A. Varshavsky, V. Ganapathy, L. Iftode, Security Versus Energy Tradeoffs in Host-based Mobile Malware Detection, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys '11, ACM, New York, NY, USA, 2011, pp. 225–238. doi:10.1145/1999995.2000017.
URL http://doi.acm.org/10.1145/1999995.2000017

[33] Heelan, Sean and Gianni, Agustin, Augmenting Vulnerability Analysis of Binary Code, in: Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC '12, ACM, New York, NY, USA, 2012, pp. 199–208. doi:10.1145/2420950.2420981.
URL http://doi.acm.org/10.1145/2420950.2420981

[34] L. Fiondella, Uncovering Weaknesses in Code With Cyclomatic Path Analysis, CrossTalk (2012) 9.

[35] S. Ouchani, O. A. Mohamed, M. Debbabi, A Security Risk Assessment Framework for SysML Activity Diagrams, in: 2013 IEEE 7th International Conference on Software Security and Reliability, 2013, pp. 227–236. doi:10.1109/SERE.2013.11.

[36] A. Bouard, J. Schanda, D. Herrscher, C. Eckert, Automotive Proxy-Based Security Architecture for CE Device Integration, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 62–76. doi:10.1007/978-3-642-36660-4_5.
URL http://dx.doi.org/10.1007/978-3-642-36660-4_5

[37] Q. Zhu, T. Başar, Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense, Springer International Publishing, Cham, 2013, pp. 246–263. doi:10.1007/978-3-319-02786-9_15.
URL http://dx.doi.org/10.1007/978-3-319-02786-9_15

[38] J. Serrano, E. Cesar, E. Heymann, B. Miller, Increasing Automated Vulnerability Assessment Accuracy on Cloud and Grid Middleware, Springer

Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 278–294. doi:10.1007/978-3-642-38033-4_20.
URL http://dx.doi.org/10.1007/978-3-642-38033-4_20

[39] Z. Han, L. Cheng, Y. Zhang, D. Feng, Measuring and Comparing the Protection Quality in Different Operating Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 642–648. doi:10.1007/978-3-642-38631-2_51.
URL http://dx.doi.org/10.1007/978-3-642-38631-2_51

[40] W. Peng, F. Li, C. T. Huang, X. Zou, A Moving-target Defense Strategy for Cloud-based Services with Heterogeneous and Dynamic Attack Surfaces, in: 2014 IEEE International Conference on Communications (ICC), 2014, pp. 804–809. doi:10.1109/ICC.2014.6883418.

[41] A. Kurmus, S. Dechand, R. Kapitza, Quantifiable Run-Time Kernel Attack Surface Reduction, Springer International Publishing, Cham, 2014, pp. 212–234. doi:10.1007/978-3-319-08509-8_12.
URL http://dx.doi.org/10.1007/978-3-319-08509-8_12

[42] E. Osterweil, D. McPherson, L. Zhang, The Shape and Size of Threats: Defining a Networked System's Attack Surface, in: 2014 IEEE 22nd International Conference on Network Protocols, 2014, pp. 636–641. doi:10.1109/ICNP.2014.101.

[43] D. Kar, F. Fang, F. Delle Fave, N. Sintov, M. Tambe, "A Game of Thrones": When Human Behavior Models Compete in Repeated Stackelberg Security Games, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2015, pp. 1381–1390.
URL http://dl.acm.org/citation.cfm?id=2772879.2773329

[44] B. Ford, T. Nguyen, M. Tambe, N. Sintov, F. D. Fave, Beware the Soothsayer: From Attack Prediction Accuracy to Predictive Reliability in Se-

curity Games, Springer International Publishing, Cham, 2015, pp. 35–56. doi:10.1007/978-3-319-25594-1_3.
URL http://dx.doi.org/10.1007/978-3-319-25594-1_3

[45] D. Bodeau, R. Graubart, W. Heinbockel, E. Laderman, Cyber Resiliency Engineering Aid - The Updated Cyber Resiliency Engineering Framework and Guidance on Applying Cyber Resiliency Techniques, Tech. Rep. MTR140499Rl, MITRE Corporation (May 2015).
URL https://www.mitre.org/sites/default/files/publications/pr-15-1334-cyber-resiliency-engineering-aid-framework-update.pdf

[46] W. Bryant, Cyberspace Resiliency: Springing Back with the Bamboo, Springer International Publishing, Cham, 2015, pp. 1–17. doi:10.1007/978-3-319-23585-1_1.
URL http://dx.doi.org/10.1007/978-3-319-23585-1_1

[47] D. Kar, F. Fang, F. Delle Fave, N. Sintov, A. Sinha, A. Galstyan, B. An, M. Tambe, Learning Bounded Rationality Models of the Adversary in Repeated Stackelberg Security Games, in: 4th annual Adaptive Learning Agents Workshop at the Conference on Autonomous Agents and Multiagent Systems (ALA-AAMAS), 2015.

[48] T. UcedaVélez, M. Morana, Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis, Wiley, 2015.
URL https://books.google.com/books?id=pHtXCQAAQBAJ

[49] U. F. C. Command, U. T. Fleet, C. Magazine, U.S. Fleet Cyber Command/TENTH Fleet Strategic Plan 2015 2020, Tech. rep., U.S. Fleet Cyber Command / TENTH Fleet (2015).

[50] S. M. Bellovin, Attack Surfaces, IEEE Security Privacy 14 (3) (2016) 88–88. doi:10.1109/MSP.2016.55.

[51] S. Wheatley, T. Maillart, D. Sornette, The extreme risk of personal data breaches and the erosion of privacy, The European Physical Journal B 89 (1) (2016) 7. doi:10.1140/epjb/e2015-60754-4.
URL http://dx.doi.org/10.1140/epjb/e2015-60754-4

[52] P. Manadhata, J. M. Wing, An Attack Surface Metric, Tech. Rep. CMU-CS-05-155, Carnegie Mellon University (July 2005).
URL http://reports-archive.adm.cs.cmu.edu/anon/2005/CMU-CS-05-155.pdf

[53] P. K. Manadhata, J. M. Wing, An Attack Surface Metric, in: Proceedings of First Workshop on Security Metrics (MetriCon), 2006.

[54] M. Howard, S. Lipner, The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software, Vol. 8, Microsoft Press Redmond, 2006.

[55] P. K. Manadhata, Y. Karabulut, J. M. Wing, Measuring the Attack Surfaces of SAP Business Applications, Tech. Rep. CMU-CS-08-134, Carnegie Mellon University (May 2008).
URL http://reports-archive.adm.cs.cmu.edu/anon/2008/CMU-CS-08-134.pdf

[56] T. Heumann, S. Türpe, J. Keller, Quantifying the Attack Surface of a Web Application., in: Sicherheit, 2010, pp. 305–316.
URL http://publica.fraunhofer.de/documents/N-142624.html

[57] E. Chin, A. P. Felt, K. Greenwood, D. Wagner, Analyzing Inter-application Communication in Android, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys '11, ACM, New York, NY, USA, 2011, pp. 239–252. doi:10.1145/1999995.2000018.
URL http://doi.acm.org/10.1145/1999995.2000018

[58] J. Bird, J. Manico, Attack Surface Analysis Cheat Sheet, https:

//www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet (July 2015).

[59] P. K. Manadhata, J. M. Wing, A Formal Model for a System's Attack Surface, Springer New York, New York, NY, 2011, pp. 1–28. doi:10.1007/978-1-4614-0977-9_1.
URL http://dx.doi.org/10.1007/978-1-4614-0977-9_1

[60] B. Martin, S. C. Coley, Common Weakness Scoring System (CWSS), http://cwe.mitre.org/cwss/ (Sep 2014).

[61] S. Northcutt, The attack surface problem, SANS Technology Institute Document.
URL https://www.sans.edu/cyber-research/security-laboratory/article/did-attack-surface

[62] E. Knapp, R. Samani, Applied Cyber Security and the Smart Grid: Implementing Security Controls into the Modern Power Infrastructure, Elsevier Science, 2013.
URL https://books.google.com/books?id=_9GzAzehLLUC

[63] J. Kasten, E. Wustrow, J. A. Halderman, CAge: Taming Certificate Authorities by Inferring Restricted Scopes, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 329–337. doi:10.1007/978-3-642-39884-1_28.
URL http://dx.doi.org/10.1007/978-3-642-39884-1_28

[64] N. Gruschka, M. Jensen, Attack surfaces: A taxonomy for attacks on cloud services, in: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, IEEE, 2010, pp. 276–279.

Appendix A. Quasi-gold Standard Set

The studies listed below composed the quasi-gold standard set used in the validation of search keywords in the selection stage of the SLR.

Q1 Approximating Attack Surfaces with Stack Traces [7]

Q2 Beyond the Attack Surface: Assessing Security Risk with Random Walks on Call Graphs [6]

Q3 Measuring the Attack Surfaces of Two FTP Daemons [18]

Q4 Report: Measuring the Attack Surfaces of Enterprise Software [19]

Q5 An Attack Surface Metric [17]

Q6 An Attack Surface Metric [20]

Q7 An Approach to Measuring a System's Attack Surface [21]

Q8 Measuring a System's Attack Surface [4]

Q9 Measuring Relative Attack Surfaces [12]

Q10 Measuring Relative Attack Surfaces [22]

Q11 Using Attack Surface Entry Points and Reachability Analysis to Assess the Risk of Software Vulnerability Exploitability [5]

Q12 Using Software Structure to Predict Vulnerability Exploitation Potential [15]

Q13 Relationship between Attack Surface and Vulnerability Density: A Case Study on Apache HTTP Server [23]

Q14 Improving Software Security by Identifying and Securing Paths Linking Attack Surface to Attack Target [24]

Q15 Automatically Securing Permission-based Software by Reducing the Attack Surface: An Application to Android [25]

Q16 k-Zero Day Safety: Measuring the Security Risk of Networks against Unknown Attacks [26]

Q17 Cyber Attack Exposure Evaluation Framework for the Smart Grid [27]

Appendix B. Attack Surface Definition Papers

The papers listed below define, formally or informally, the phrase *attack surface*.

D1 Measuring Relative Attack Surfaces [22]

D2 Security benchmarks of OSGi platforms: toward Hardened OSGi [28]

D3 User Security [29]

D4 A Formal Model for a System's Attack Surface [30]

D5 Introducing Diversity and Uncertainty to Create Moving Attack Surfaces for Web Services [31]

D6 Security Versus Energy Tradeoffs in Host-based Mobile Malware Detection [32]

D7 Augmenting Vulnerability Analysis of Binary Code [33]

D8 Uncovering Weaknesses in Code With Cyclomatic Path Analysis [34]

D9 A Security Risk Assessment Framework for SysML Activity Diagrams [35]

D10 Automotive Proxy-Based Security Architecture for CE Device Integration [36]

D11 Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense [37]

D12 Increasing Automated Vulnerability Assessment Accuracy on Cloud and Grid Middleware [38]

D13 Measuring and Comparing the Protection Quality in Different Operating Systems [39]

D14 A Moving-target Defense Strategy for Cloud-based Services with Heterogeneous and Dynamic Attack Surfaces [40]

D15 Quantifiable Run-Time Kernel Attack Surface Reduction [41]

D16 The Shape and Size of Threats: Defining a Networked System's Attack Surface [42]

D17 "A Game of Thrones": When Human Behavior Models Compete in Repeated Stackelberg Security Games [43]

D18 Beware the Soothsayer: From Attack Prediction Accuracy to Predictive Reliability in Security Games [44]

D19 Cyber Resiliency Engineering Aid - The Updated Cyber Resiliency Engineering Framework and Guidance on Applying Cyber Resiliency Techniques [45]

D20 Cyberspace Resiliency: Springing Back with the Bamboo [46]

D21 Learning Bounded Rationality Models of the Adversary in Repeated Stackelberg Security Games [47]

D22 Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis [48]

D23 U.S. Fleet Cyber Command/TENTH Fleet Strategic Plan 2015 2020 [49]

D24 Attack Surfaces [50]

D25 The extreme risk of personal data breaches and the erosion of privacy [51]

D26 Attack Surface: Mitigate Security Risks by Minimizing the Code You Expose to Untrusted Users [3]

D27 Measuring a System's Attack Surface [4]

D28 Measuring Relative Attack Surfaces [12]

D29 An Attack Surface Metric [52]

D30 An Attack Surface Metric [53]

D31 Fending Off Future Attacks by Reducing Attack Surface [2]

D32 The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software [54]

D33 An Approach to Measuring A System's Attack Surface [21]

D34 Measuring the Attack Surfaces of SAP Business Applications [55]

D35 An Attack Surface Metric [17]

D36 An Attack Surface Metric [20]

D37 Writing Secure Code [13]

D38 Quantifying the Attack Surface of a Web Application. [56]

D39 Analyzing Inter-application Communication in Android [57]

D40 Attack Surface Analysis Cheat Sheet [58]

D41 Report: Measuring the Attack Surfaces of Enterprise Software [19]

D42 A Formal Model for a System's Attack Surface [59]

D43 Common Weakness Scoring System (CWSS) [60]

D44 The attack surface problem [61]

D45 Applied Cyber Security and the Smart Grid: Implementing Security Controls into the Modern Power Infrastructure [62]

D46 CAge: Taming Certificate Authorities by Inferring Restricted Scopes [63]

D47 Attack surfaces: A taxonomy for attacks on cloud services [64]

Appendix C. Themes of Cited Definitions

Shown in Table C.1 is a list of papers that define, formally or informally, the phrase *attack surface*, organized based on conceptual similarity.

Table C.1: List of definitions with at least one citation in their associated themes

Paper ID	Definition	# Citations
	Methods Theme	
D28	We describe a system's attack surface along three abstract dimensions: targets and enablers, channels and protocols, and access rights.	25
D1	We describe a system's attack surface along three abstract dimensions: targets and enablers, channels and protocols, and access rights.	13
D27	We define an attack surface in terms of the system's actions that are externally visible to its users and the system's resources that each action accesses or modifies.	11
D29	We define the attack surface of a system in terms of the system's attackability along three abstract dimensions: method, data, and channel.	10

Continued on next page

Table C.1. List of definitions with at least one citation in their associated themes (Continued)

Paper ID	Definition	# Citations
D4	We formalize the notion of a system's attack surface using an I/O automata model of the system and define a quantitative measure of the attack surface in terms of three kinds of resources used in attacks on the system: methods, channels, and data.	3
D34	Intuitively, a system's attack surface is the subset of the system's resources (methods, channels, and data) used in attacks on the system.	1
D38	The attack surface of a system represents the exposure of application objects to attackers and is affected primarily by architecture and design decisions.	1
Adversaries Theme		
D35	Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage.	43
D33	Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage.	6

Continued on next page

Table C.1. List of definitions with at least one citation in their associated themes (Continued)

Paper ID	Definition	# Citations
D36	In this thesis, we formalize the notion of a system's attack surface and use the measure of a system's attack surface as an indicator of the system's security. Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage.	4
D32	Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage.	4
D42	Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage.	3
D30	Intuitively, a system's attack surface is the set of ways in which an adversary can attack the system.	2
	Flows Theme	
D26	The attack surface of an app is the union of code, interfaces, services, protocols, and practices available to all users, with a strong focus on what is accessible to unauthenticated users.	6

Continued on next page

Table C.1. List of definitions with at least one citation in their associated themes (Continued)

Paper ID	Definition	# Citations
D40	The Attack Surface of an application is: the sum of all paths for data/commands into and out of the application, and the code that protects these paths (including resource connection and authentication, authorization, activity logging, data validation and encoding), and all valuable data used in the application, including secrets and keys, intellectual property, critical business data, personal data and PII, and the code that protects these data (including encryption and checksums, access auditing, and data integrity and operational security controls).	2
Features Theme		

Continued on next page

Table C.1. List of definitions with at least one citation in their associated themes (Continued)

Paper ID	Definition	# Citations
D31	List of attack features: Open sockets, Open RPC endpoints, Open named pipes, Services, Services running by default, Services running as SYSTEM, Active Web handlers (ASP files, HTR files, and so on), Active ISAPI Filters, Dynamic Web pages (ASP and such), Executable virtual directories, Enabled Accounts, Enabled Accounts in admin group, Null Sessions to pipes and shares, Guest account enabled, Weak ACLs in the file system, Weak ACLs in Registry, Weak ACLs on shares	12
D37	List of attack features: Open sockets, Open RPC endpoints, Open named pipes, Services, Services running by default, Services running as SYSTEM, Active Web handlers (ASP files, HTR files, and so on), Active ISAPI Filters, Dynamic Web pages (ASP and such), Executable virtual directories, Enabled Accounts, Enabled Accounts in admin group, Null Sessions to pipes and shares, Guest account enabled, Weak ACLs in the file system, Weak ACLs in Registry, Weak ACLs on shares	1
Barriers Theme		

Continued on next page

Table C.1. List of definitions with at least one citation in their associated themes (Continued)

Paper ID	Definition	# Citations
D43	Attack Surface metric group: the barriers that an attacker must overcome in order to exploit the weakness.	1
	Reachable Vulnerabilities Theme	
D44	We can define attack surface as our exposure, the reachable and exploitable vulnerabilities that we have.	2