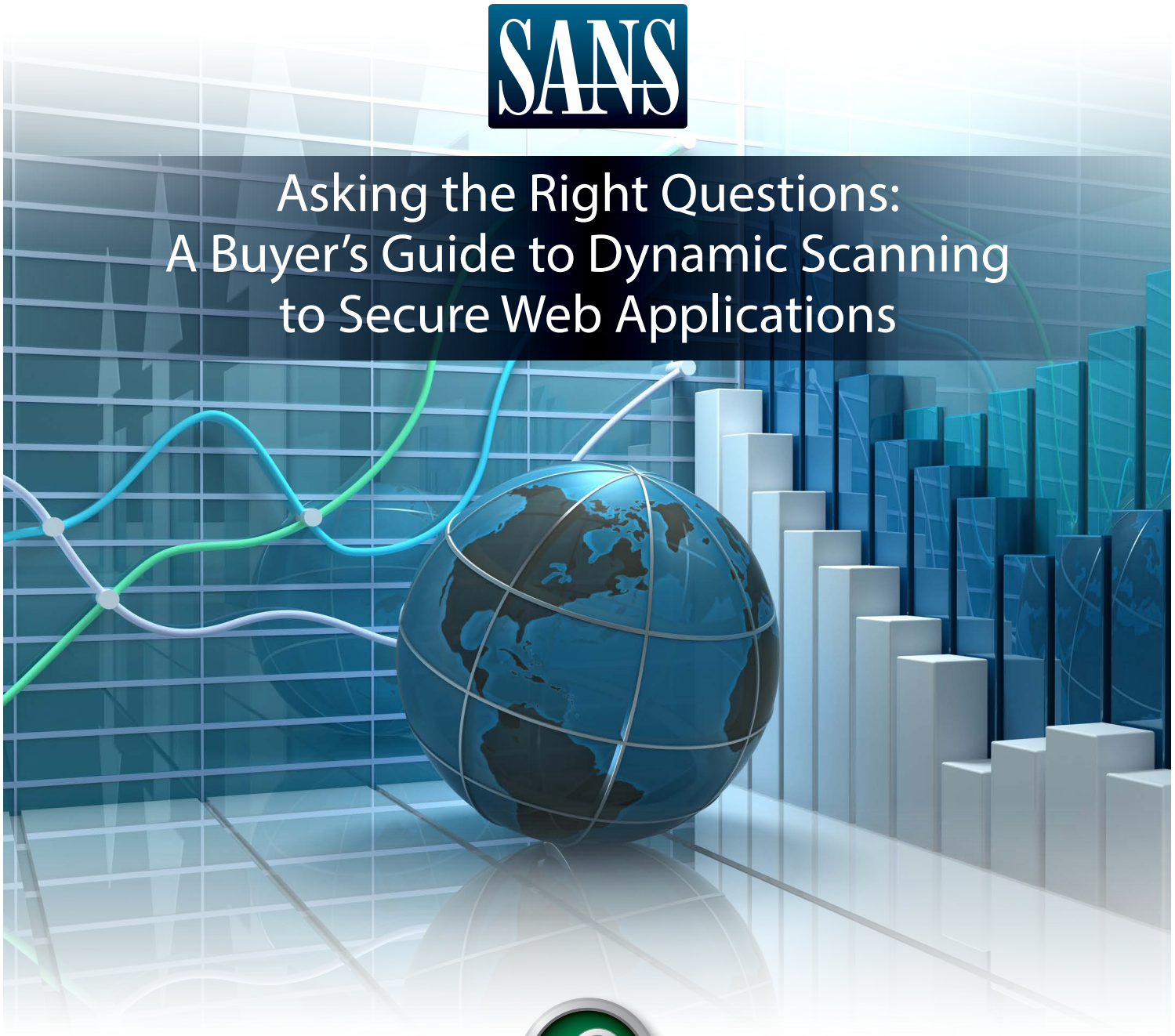
Secure Coding.
Practical steps to defend your web apps.



Asking the Right Questions: A Buyer's Guide to Dynamic Scanning to Secure Web Applications

A SANS Whitepaper

Written by Barbara Filkins

September 2017

Sponsored by
Veracode

Getting Started

Once an organization has made the decision to invest in dynamic application security testing (DAST) to support its application security program—potentially across all phases of the software development life cycle (SDLC), including production—the next challenge becomes how to proceed. What is the best process to determine and procure what is really needed?

For this reason, SANS has developed a buyer's guide for procurement of a DAST solution, whether as a product or software-as-a-service (SaaS). This guide will lay out an effective process to evaluate, select and implement the best solution that can be followed by any organization, large or small.

This guide provides a four-step method, shown in Figure 1, for acquiring the solution you need to enable your organization's use of DAST.

Figure 1. Four-Step Process to DAST Procurement

Step 1: Identify—Understand What You Need

Selection of a DAST solution starts with a certain amount of internal due diligence. This diligence is needed to fully understand the goals and objectives of your organization as related to DAST, both to establish the value proposition of DAST to your organization and to clearly and concisely communicate the required background and requirements to your potential vendors.

To get the right kind of support, the lead analyst on a procurement team needs to:

- Understand what is needed for effective use of the solution within your enterprise and communicate those needs to your internal stakeholders. Getting commitment for the budget, internal resources and schedule to integrate DAST into existing or new workflows is needed for successful fielding, not just procurement, of the solution.
- Develop the high-level requirements the DAST solution must meet.
- Gather background information to help vendors size their response and to assist your team in evaluating the offers.
- Conduct a high-level market survey, if needed, to gain better insight into your procurement approach before actually publishing a request for proposals (RFP).

Getting commitment for the budget, internal resources and schedule to integrate DAST into existing or new workflows is needed for successful fielding, not just procurement, of the solution.

Don't forget to look at both functional and risk-driven requirements while trying to evaluate the high-level requirements behind your use of DAST.

Functional testing focuses on whether the tool does what it is supposed to do, whether it behaves as expected in specific situations and how specific security functions work—such as authentication, authorization or encryption. Risk-driven assessments ask not what a tool does, but whether it does things you don't want it to do. Use risk analysis and threat modeling to establish possible adverse scenarios and related test requirements that demonstrate how well, under realistic circumstances, a tool you're considering is able to identify exploitable weaknesses and mitigate potential threats.

Don't neglect the value of using DAST after an application is deployed to production. Post-implementation dynamic testing can help identify errors in configuration, identify the potential for exploitation in correct configuration plans and identify previously invisible problems that might appear once an application has been released into an open environment.

Next, gather and present background information on your application infrastructure to clarify the scope and magnitude of the solution you need and to provide enough information to give prospective suppliers the ability to address the environment in which your applications live, not just the one in which a potential solution has been developed or tested.

A short checklist of the items you need to collect and why each is important is shown in Table 1.

Table 1. Application Inventory: What You Need to Know

Item	Why
Application inventory, including: • Number and types of apps • Platforms built upon (e.g., Apache) • Underlying infrastructure, including operating systems and utilities, hardware and software servers, use of virtualization	Understand the types of web applications you might expect to encounter in using DAST. Legacy apps, for example, may have old or outdated versions of software components with serious and unsuspected operational vulnerabilities that are begging to be exploited, while current apps rely heavily on client-side components that may also be easily exploited. Establish budget based on sizing metrics. Communicate magnitude of service or scalability of product to vendor.
Expected users, including number and roles (e.g., developers, quality assurance and security)	Gather usability expectations and requirements in terms of dashboard and reports. Identify licensing structure (if by user). Determine skill level of users and the need for vendor support services, such as developer assistance.
SDLC methodology used for application development and testing	Identify tools or services that integrate with current application development and testing platforms (to avoid product mismatch). Determine level of automation required to support existing workflows or expected if new workflows are envisioned. Establish timing and frequency of testing. Create service-level timeline.
Number and types of applications to be tested (minimum to maximum)	Project number of licenses needed for product or size of testing pool for SaaS contract.

A high-level market study of DAST solutions can be used to solicit vendor interest as well as help you gain a better understanding of what DAST products and services are available.

Sources to identify potential vendors can include:

- Hands-on experience of current staff with specific DAST products and services
- Previous projects such as reviews of requirements and analytical methods used
- Trade journals and newsletters specific to the type of technology or application
- Internet searches
- Professional association memberships
- Trade shows such as RSA
- Vendor literature library
- Special interest groups
- Vendor user groups
- Independent technology assessment reports, such as those from Gartner or other market analyst firms

Ask each potential vendor to submit product information that matches the high-level requirements you have developed. The questions you ask can be fairly broad. You simply have to say that your organization is seeking products or services able to:

- Operate a dynamic application security testing (DAST) platform (or SaaS) that can cover and detect new types of vulnerabilities.
- Prioritize security vulnerabilities in running applications and interactively communicate data to reproduce and fix discovered issues.
- Provide comprehensive details about the vulnerability detected.
- Identify, prioritize and validate critical, high-risk security vulnerabilities.
- Comply with internal and external security policies and regulations, giving references appropriate to your industry.
- Monitor applications on a regular basis for changes in the security posture.
- Generate customized management or compliance reports.
- Support the latest web technologies—at minimum, HTML5, JSON, AJAX and JavaScript.
- Test mobile-optimized websites or web-enabled service interfaces.
- Provide dashboard management visibility to track vulnerabilities, confirm remediation and review trends.
- Provide a validated method to reduce false positives and negatives.

As you evaluate the results from your study, maintain a distinction between what your actual requirements will be and those product features that would be nice to have, but which are not essential.

Benefits of a Market Study

- Integrates your high-level requirements with existing market capabilities.
- Informs your decision and evaluation criteria for procurement.
- Streamlines the actual selection and evaluation process by identifying those vendors that may have a potential solution.
- Reveals potential risks related to confidentiality and other legal conditions related to your application portfolio.
- Provides background information on the responding vendors, including organizational information (e.g., company name and address, points of contact and recent experience on similar projects) that can be used to build a vendor list for the formal request for proposals.

The outcome of this first step is to create as complete an understanding as possible of what your organization needs. That is, you do not simply need to procure a DAST solution; you also need to implement it within your organization. Thorough, up-front preparation will help ensure that your organization will be satisfied with its investment and able to use it wisely!

Step 2: Strategize How to Approach Procurement

You have collected a lot of information in the previous step. Now, you need to synthesize this and develop your procurement documents—including how you will evaluate vendor responses—and another document defining the approach you want to take for selecting the vendor.

You need to establish your formal requirements for the solution and how you will evaluate how a vendor responds to each.

The following are four major sets of factors to examine when deciding which solutions and vendors to consider:

1. First, there are the specific *product/service requirements* that define the actual solution. These are the technical and functional requirements divided into three subcategories:
   - **SDLC functionality and support**—enumerates what the solution actually must do, including what security vulnerabilities it must detect and help remediate, what functions it should support to enable automated workflow and test support, etc.
   - **Analysis and reporting**—provides the methods for risk rating, based on standard and proprietary scales, and the dashboard and reports that provide visibility into the testing and remediation processes.
   - **Product/service environment**—needs to account for integration into the existing infrastructure of an enterprise as well as scalability and ease of use.
2. The second major set of variables to nail down involves the level of support or supporting services that may be required, especially if the in-house staff's expertise in application security is limited, or if the support or information services to be used to research vulnerabilities are limited based on technical, organizational or cost issues.
3. The third set of requirements involves the performance histories and backgrounds of the vendors bidding for the contract. Origin and past performance aren't perfect predictors of behavior, but can provide a good indication of how well a potential supplier will be able to meet the support needs of your organization. (See Table 2 later in this paper.)
4. The last set of criteria is often the first for non-technologists on the procurement team: *price comparisons*. Most vendors offer prices structured according to how the product or service is configured. The challenge here will be acquiring enough information about each vendor's pricing structure to create a realistic pricing model, especially if you need an apples-to-apples comparison of the prices of two or more vendors.

Be Smart About Evaluating Cost

Be careful when evaluating cost. Don't be fooled into selecting a vendor simply because its price is the lowest. You want to know total cost, not the sticker price. Be sure to ask about factors that could inflate a low initial cost, including the following:

- How many hours of training will be required for each operator? Who will provide the training, and at what hourly rate? If the training is off-site, what travel costs are involved, and is distance learning an option to keep those costs low?
- How many hours would it take to fully configure each solution being considered until it meets your needs? What is the recurring configuration cost when you move to a new set of applications? Reconfiguration costs and delays may be bigger problems for agile or DevOps shops that are likely to redeploy apps more frequently.
- How long will it take for your staff to become fully proficient on the solution, and what will be the cost savings (if any) resulting from the use of improved automation?

Product/Service Requirements: Where to Start?

Determining what your organization actually needs to accomplish with its DAST solution, and how to prioritize those requirements, may be the most interesting part of the evaluation, but it can also be the most challenging. How complex should you make the requirements? How explicitly should you state precisely what is needed? SANS has provided evaluation matrices as a starting point to help you develop your own RFP, but it is worth highlighting a few issues here:

- **Accuracy.** Vulnerability and application security testing is a critical issue. The greater the accuracy, the lower the number of false-positive alerts that have to be investigated or remediated by developers. High false-positive rates reduce developer productivity by forcing them to chase bugs that may not exist—which can derail an AppSec program because developers no longer trust the tool or process.

  Bear in mind that no test is flawless at identifying vulnerabilities. Alerts have to be investigated, often by subject-matter experts who are not only fewer in number but more highly paid than most of the rest of your staff. This could turn into another hidden cost if you lack such gurus on staff, or if the number of legitimate vulnerabilities to fix is great. Hiring temporary or part-time contractors with the appropriate expertise could bring those costs down, but whether it makes financial sense depends on the number of alerts, their complexity and the hourly rates of contractors with the specific types of expertise you need.

Efficient use of time is important, but don't be fooled into using the speed with which a DAST solution sifts through websites and applications as the primary point of comparison.

Speed is important, but the time cost associated with investigating false positives means that accuracy will trump speed every time.

The need to keep pace with DevOps or continuous development teams that churn out code constantly may not be as critical an issue as it seems. DevOps teams may automate their pipeline but may not decide to run DAST against every build. They may, for example, decide to run DAST on a specific schedule or once the application has been deployed to a staging environment.

The worst-case scenario in a continuous development model would be that a vulnerability may be out, live on the web for a few hours. Letting low-severity bugs live for a few hours may be considered an acceptable business risk, as long as the team commits to fixing high-severity vulnerabilities before release.

- **Authentication.** Configuring authentication can also be an issue for some scanning solutions, which may be less capable of automatically verifying whether a login process has been successful. A problematic login by a DAST solution can result in less coverage. Some methods—for example, particularly heavy, multistep JavaScript login forms—can be especially difficult to configure for and test. It can take hours to get the configuration right when the process is tricky, even for application security experts who have a deep understanding of the authentication mechanism.

  Consider vendors with operational teams that have demonstrated they can help figure out authentication challenges that may come up as part of the scan. Vendors with their own expertise can give your security team the chance to spend more time focused on its main job, in addition to making it more likely scans will be started and completed on schedule. That particular capability may be available only from vendors offering SaaS-based solutions, which may be a factor when you try to decide between on-premises or cloud-based scans.

- **Scanning.** When scanning production environments, companies can run the risk of configuring their scanner to be too aggressive, which can either impact application performance or even bring it down. Some vendors support production-safe scanning with settings that ensure a scan never brings down your production environment. Another upside of this functionality is that you are no longer restricted to scanning windows outside of core business hours. Be sure to ask a potential vendor how it ensures production-safe scanning.

To SaaS or Not to SaaS?

Should you use SaaS or on-premises tools? Ultimately, this boils down to a basic question: Do you have the resources—staff and tools—needed to deploy and manage the application security program you need, or would it be simpler to outsource those functions to a service provider?

As a first step, consider hardware expenses, including initial investment, deployment and maintenance. The difference in performance can be stark if you compare a single-server on-premises solution with the clustered, high-performance hardware and multiprocessing software typical of SaaS implementations designed to serve many customers at once. Many on-premises solutions are limited to one concurrent scan (i.e., single threaded). Raising performance by running several applications in parallel could mean spending additional dollars on a high-performing cluster or server farm.

SaaS providers have cost-performance issues, too. Some limit the number of parallel scans available from their platform, so it is important to know what level of concurrency to expect.

At the other end of the spectrum, in terms of scale, are highly specialized tools designed to give individual consultants high levels of control over performance and customization, which may help them troubleshoot individual problems more quickly than an InfoSec staff executing a broad-based security program.

Scanners based on laptops often have difficulty trying to scale to more powerful hardware. However, the tools designed for consultants tend to require considerable expertise and manual intervention for both scans and remediation, skills that may be in short supply at many organizations.

On the other hand, a SaaS vendor may offer highly scalable, automated DAST scanning with service components that minimize the effort and required expertise on your side. When comparing SaaS versus product solutions, consider carefully who on your team will have the time and expertise to run these solutions and how often they would be able to carry out the audits. This can serve as a good check as to the approach your organization should take.

Support Services

Think about the roles you will need to support. Your AppSec assessment team will probably include architects, developers or other testing specialists who were not involved in the app's development. If these resources are lacking or too busy, consider third-party validators or service providers. And if you do augment your internal team with vendor resources, what do you need from the service provider, and how will it work with your internal capabilities?

From a developer standpoint, consider what a vendor may offer in support of remediation. When a vulnerability is found that requires code changes, your developers may not always know how to fix it. This will require considerable research time for your organization's developers. Many companies don't have the bandwidth to help developers, and few have the combined software development and application security background required to do this job well.

If this scenario is representative of your organization, consider vendor services where developers can schedule a call with an application security expert who has a development background to review the problem and discuss options. These types of services can dramatically increase fixed rates and, therefore, the success of the program.

See Table 2 for requirements, objectives and criteria on which to evaluate AppSec security service vendors.

Table 2. Strategy for AppSec Security Services

Requirement	Objective	Evaluation Questions/Criteria
Provide training support	Define what you need training for (development, testing, specific tools) and who needs the training. Then, assess the quality of vendor-provided training against these requirements.	Is the vendor willing to provide knowledge transfer, suggestions or recommendations that improve your AppSec processes? Does the vendor provide feedback on student progress? Does the vendor deliver training in the manner(s) required by your organization, such as on-site, on-demand, or self-paced (automated)? Does the vendor provide training materials that your organization can license for further use?
Provide application security consulting and operational assistance	Define whether you have adequate application security expertise to properly advise all your development teams.	Does the vendor offer application security consulting, and if so, how is it priced (e.g., by the hour or as a program)? Can you justify any cost savings or better response times resulting from your developers having timely access to an AppSec expert? Does the vendor offer services that will help with DAST configuration? Does the vendor offer vulnerability verification services that remove false positives from reports?
Participate in SDLC processes	Evaluate vendor experience across the application lifecycle—SDLC to production.	Do you need to cover both production and SDLC use cases? If so, is the vendor being considered comfortable supporting both SDLC and production use cases? Will the vendor combine all testing results in one policy pass/fail, saving time for compliance reporting? Will the vendor include manual testing results? Can the vendor provide program management services for your application security program, if desired?

Vendor Background Due Diligence in a Nutshell

Evaluating vendor background helps guard against the selection of a vendor with known service performance issues. You are looking for a vendor that provides products and services that align with its own stated high standards and which demonstrates sufficient service capacity and business stability to meet your enterprise needs. You can also identify (and hopefully) remove any hidden cost drivers.

Considerations for evaluating a vendor's background are outlined in Table 3.

Table 3. Strategy for Due Diligence as Outlined in Vendor Procurement Document

Item	Purpose/Objective	Evaluation Criteria
Verify past performance	Evaluate the likelihood a vendor will meet the expectations for support defined in the final agreement.	• Does the vendor have a good history of performance? • Has the vendor been successful in providing tools and services to organizations of your size and industry type? • Can the vendor provide references? Try for at least three verifiable references from projects similar in size and scope to that planned by your organization.
Verify financial stability	Evaluate the stability of the vendor in order to establish the vendor's long-term availability to work with your organization.	• How long has the vendor been in business? • Is it expanding, in status quo mode, or consolidating or being acquired? • Does an online search (e.g., Google) reveal any potential red flags in the business media? • Does a Dun & Bradstreet report on the firm verify information?
Assess contract terms	Evaluate the contract to ensure that the statement of work (SOW) requirements align with the contract.	• What are the service level agreements for response time for support or responding to a service request? • Are there liquidated damages associated with the contract? • Is there a termination clause to the contract? • What are the intellectual property protections?
Evaluate vendor service road map	Evaluate the vendor's flexibility to meet new demands, configuration, language and other issues.	• Does the vendor perform background checks on its employees and contractors? • Does the vendor have a clear vision of where the DAST application security market is headed, including assessment methods and technology? • Is there evidence that it actively supports this vision through a strategic road map that includes investment in staff training, continuous improvement of current methods, and participation in key conferences?

Step 3: Gather and Evaluate

With requirements established and a handle on evaluation, you are ready to publish an RFP to the vendor community at large or to a preselected pool of vendors.

The major evaluation steps are shown in Table 4.

Table 4. Major Evaluation Steps

Step	Contents
Document procurement objective, scope and background. Note: Consider how much information you want to reveal to the vendor at this point. Some of this information may be sensitive.	1. Describe the overall purpose of your DAST project, based on your assessment of what you need. 2. Define the scope. Are you acquiring a DAST product, service or both? 3. Describe your application development methodology and whether you have a current AppSec program. 4. Define your users. 5. Describe your environment, including your application inventory and a breakdown of the number of types of applications you expect to use and install. 6. Do you have a compliance objective? Internal, external or both? Which regulations (e.g., HIPAA, SOX) apply?
Prepare the vendor response tools and your approach. Note: Review the accompanying matrix to this guide.	1. Decide what requirements are mandatory versus optional. (See the columns in the accompanying matrix.) 2. Establish your evaluation scoring method for the written response: • First pass: Pass/fail criteria ✔ Did the vendor complete the organizational information? (Section 1.0) ✔ Did the vendor address all the mandatory requirements? (Section 3.0) • Second pass: Evaluation and scoring of Solution Overview ✔ Assess point value from 0 to 50 for overall section. ✔ Assess based on completeness of response and perceived risk. ✔ Evaluate the scenario. Did the example meet your expectations as to how the solution might perform in your enterprise? • Third pass: Evaluation and scoring of Support Services ✔ Assess point value from 0 to 50 for overall section. ✔ Assess based on completeness of response. ✔ Evaluate the response. Does the vendor support approach meet the needs of your enterprise? • Fourth pass: Evaluation of Pricing and Components ✔ Does the final configuration of the solution meet your established budget?
Evaluate the vendor responses and select top vendors for further consideration.	Compare the top scores and either select the top vendor or two or more for a comparison. Comparison can be executed through 1) scripted demonstrations, 2) site visits to vendor client(s), or 3) proof of concept. The latter option is perhaps the most valuable comparison, but it will also take the most resources if done correctly.

Conducting a Proof of Concept

You undoubtedly have questions from what you have discovered in the "Steps" evaluation process previously outlined: identify, strategize and evaluate. Depending on your potential investment, you may want to conduct a proof of concept (POC). Here are the issues you will need to consider to execute a successful POC:

- **Know your goals.** You will use the POC to become educated or familiar with the solution. You want to be certain the solution meets your operational guidelines. You also want to discover whether there are any additional hidden costs, such as the need for new skill sets in your personnel or additional services that you should purchase from a vendor.
- **Define the POC.** Work with your vendor to define the POC such that it meets your expectations. The vendor should clearly outline the process it intends to follow and how it plans to address your organization's unique challenges—that is, which of your requirements can be fully met or not met in your environment. This conversation will also give you a sense of how the partnership will unfold if you select this vendor.
- **Plan to conduct the POC in your environment.** You should try to build a dedicated lab environment for your testing. It doesn't have to be a full-blown replica of your production environment—nor even of your full application development and testing environment. But you should know how it would scale to meet either the production or full testing environments.
- **Failing to plan is planning to fail.** Determine the timeline you consider sufficient for POC testing. There are no shortcuts to a successful POC. And don't rush it. The more it is rushed, the more likely it is to fail. This is a huge investment of time and labor for both you and the vendor.
- **Conduct the POC.** Ask questions, and document everything! Include the steps of installation, results of tests that were performed, and final findings and recommendations. Consult any checklists you may have to ensure you have not forgotten anything. By knowing the limitations of your POC environment, for example, you can identify possible issues you might encounter when moving the DAST solution into the full test or production environments, such as realizing performance limitations before it's too late, avoiding procurement of the wrong feature set or device, and underestimating resources for proper configuration and deployment.

Step 4: Awarding the Contract

The final step is anticlimactic, but far more work than the others. Once you've completed the process of information gathering, evaluation, and communication with stakeholders, vendors and employees, and have come to a conclusion about the DAST solution that is the best fit for your organization, the only thing left to do is to move forward.

Following the process described in this white paper should provide you with the information and insight needed to help you make a clear, defensible procurement decision. It should provide your organization with sufficient opportunity to review contractual documents, service level agreements and collateral such that there should be no surprises during the final negotiation and execution of a contract.

You will need to fully engage your internal counsel for these final steps, but if you've kept them up to date with any legal considerations and contractual concerns during the evaluation, their involvement should not add significant delays. If you've also kept non-technical stakeholders, especially senior management, up to date, discussions and decisions about budgets and ongoing funding for training and implementation should also have been resolved.

The steps described here offer a framework for evaluating and making a complex decision. However, they don't do anything more than help you prepare to improve your application security or overall security stance.

Choosing the right tools and right approach to application scanning and security is a critical step for any organization, but making it happen requires leadership, not just research and analytical skills. Fortunately, if your organization has the interaction, communication, negotiation and management skills necessary to complete this process successfully, that is a good indication it has the kinds of skills needed to complete the implementation as well.

Good luck.

Request for Proposal (RFP) | Request for Information (RFI) Checklist for DAST Testing Product Vendor Selection Version 1.0

September 12, 2017

General Information

Our organization, [insert name of organization seeking the DAST solution (or "Organization")], seeks a DAST solution, whether products, software-as-a-service (SaaS), or a hybrid, that can provide uniform standards for the entire enterprise. We seek a solution that combines the advantages of best-of-breed technology with the benefits of an integrated, scalable and manageable solution. Our goal is to acquire and implement a robust enterprise solution that is flexible enough to support individual user and device needs but also provides effective and efficient centralized administration, management and recovery.

Organization Description

The following section describes our organization so you, the vendor, can get a feel for our deployment scenario and use this information to establish a basis for your response to Section 5.0, Pricing and Components. [Insert brief description of "Organization," including type of business, geographic placement of offices, high-level network topology, number and type of web-based applications, development methodology followed, number of users and any other information that may be relevant to the vendor.]

Contact Information

[Provide contact information for the RFP.]

Vendor Response

Please complete the following sections according to the instructions in each section.
[Review requirements, tailor or update areas indicated in blue, and determine whether requirement is "required" (i.e., mandatory) or "desired." Note: In this RFP checklist, "desired" DOES NOT mean the same as "optional," as a configuration option can be a "mandatory" requirement.]

1.0 Organization Background

1.0	Organizational Background
1.1	Provide pertinent contact information for your business, location of headquarters and major field offices.
1.2	Provide overall statement of revenue with breakdown as follows: • Percent attributable to DAST product(s). • Percent attributable to DAST services.
1.3	Provide total number of years in business with specific details: • Number of years providing DAST product(s). • Number of years providing DAST services.
1.4	Describe your DAST customer base: • DAST Product(s): Number of customers, number of user licenses, industries. • DAST Services: Number of customers, number of websites tested via the service, industries. Note: Provide your definition of a website (e.g., specific URLs).
1.5	Indicate your industry involvement, such as membership in industry organizations, participation in standards bodies and so forth. Provide a list of your solution partners.
1.6	Do you work with regional partners or value-added resellers for regional expertise and implementation support? If so, please provide your list.
1.7	What is your perception of market direction, and how does this affect your technology road map? Describe your anticipation of industry and customer trends, how your product plans will meet these trends, and your approach to ensure that your solution can adapt and improve while continuing to provide value to an existing customer base.

2.0 Solution Overview

Provide a comprehensive overview of your product(s)/services and how you can meet our specific requirements as provided in Section 3.0, addressing the specific concerns for each area in your narrative as outlined below.

Section	Requested Topics
2.1 SDLC Support and Functionality	Summarize the features of your DAST solution to include the specific requirements in Section 3.0. Describe any testing capability of your DAST offering not covered in the section above. Consider the following in developing your response to this section: 1. How does your solution support the required SDLC phases as outlined in Section 3.1.1 for methodologies our enterprise supports (e.g., waterfall, Agile, DevOps and so forth)? 2. Provide a descriptive scenario that shows your solution in action. This scenario should provide insight into the following: • Application testing that includes: - Detection of security vulnerabilities and the method(s) used - Support for multiple authentication schemes - Testing of client-side code (e.g., JavaScript) - Single-page applications - Other interfaces and protocols (e.g., use of fuzzing) - Other requirements such as malware detection • Remediation support • Assistance with login configuration • Vulnerability verification • Website discovery and prioritization • Interfacing to or direct support of asset inventory management • Approach to accuracy • Process and sources used to update vulnerabilities 3. Through this scenario, address the following: • Workflow support means that upon detection of a vulnerability, the product/service creates a "ticket" or "alert" that can be used to track remediation through to final disposition of the issue. Describe how your DAST product/service supports workflow either a) through its internal capabilities, or b) through integration with external environment. If the latter, please provide the platforms with which it integrates. • Describe how your solution supports customized automated processes using the example of an enterprise lightweight scan to identify issues, followed with more in-depth scans on assets where vulnerabilities have been identified. • How can remediation support be tailored to support our enterprise best practices for application development/testing? • What are your specific techniques to reduce false positives and negatives, and how do you ensure that the detection of vulnerabilities is accurate?
2.2 Analysis and Reporting	Summarize how your solution meets our analysis and reporting requirements; address the specific requirements in Section 3.2 and highlight any features of your product that can enhance these areas. Provide sample reports that represent the results from the descriptive scenario above as well as snapshots from your solution's dashboard. Describe in detail your rating scale and mechanism for detected vulnerabilities as well as a description of the sources used for this information.

Section	Requested Topics
2.3 Product/Service Environment	Wherever possible, our enterprise would like to leverage our investment in existing development and testing infrastructure and experience. Describe the overall architecture of your solution and its ability to support and integrate with our current environment, addressing the specific requirements in Section 3.3. In addition to the named application development and/or testing environments in Section 3.3, provide any additional platforms with which your solution integrates. Describe your support for web application firewalls (WAF), including those you support as part of your basic offering. How will this knowledge of web-based vulnerabilities be used to automatically test for WAF protection? Please indicate which vendors you can support, including cloud-based vendors. Describe how your solution integrates with identity and access management (IAM) systems for real-time authentication during tests. Indicate the common IAM systems with which your product/service integrates. Note: SaaS vendors, list what tools you are using as part of your offering. Product vendors, describe hardware requirements. Describe what you consider "minimal disruption" and how your solution achieves this goal. Describe how your product/service scales. Reference current customers/installations of similar size, scope and complexity to our organization. Provide at least three customer references. What is the largest application your product/service has been used to test? Does your product/service automatically maintain performance with increased workload? In general, what do you consider to be your top three differentiators from competitors?

3.0 Specific Product/Service Requirements

For each requirement in Section 4.0, please provide a concise explanation of how your proposed solution will meet the specific requirement, including any additional detail requested in the requirement paragraph (e.g., requirement paragraph 4.2.1.18).

3.1 SDLC Functionality and Support

3.1.1 SDLC Support

The DAST solution shall support the following SDLC phases:		Required	Desired
3.1.1.1	Define		
3.1.1.2	Design		
3.1.1.3	Develop		
3.1.1.4	Implement/integrate		
3.1.1.5	Deploy		
3.1.1.6	Production (e.g., penetration test)		

3.1.2 Automation for Workflow and Test Support

The DAST solution shall:		Required	Desired
3.1.2.1	Create a ticket or alert upon detection of a vulnerability.		
3.1.2.2	Track disposition of the ticket internal to the tool until the vulnerability is remediated and remediation is documented (i.e., ticket closed).		
3.1.2.3	Integrate with external workflow/automation tools that accept the product/service-generated ticket and provide disposition/remediation status back to the product/service.		
3.1.2.4	Support delta analysis (i.e., compare vulnerability status at two different times on the same application).		
3.1.2.5	Provide completeness measures (e.g., be able to highlight areas of the application that were not covered by the DAST testing performed).		
3.1.2.6	Identify to the end user where a vulnerability is located (e.g., URL, sub-URL, page) depending on the user role(s).		
3.1.2.7	Support customized automated processes such as a "pre-scan" to determine a successful configuration and then follow with more in-depth scans.		
3.1.2.8	Provide a record/replay capability of vulnerabilities discovered so that the exploitation of a vulnerability can be replayed by the developer investigating the issue or later by information security to ensure the vulnerability has been addressed when retesting.		
3.1.2.9	Provide discovery of externally facing websites for prioritization in support of asset inventory procedures.		
3.1.2.10	Provide automated re-scan capability that allows re-testing of previously found issues or vulnerabilities (e.g., in support of regression testing).		

3.1.3 Remediation Support

The DAST solution shall:		Required	Desired
3.1.3.1	Provide remediation advice (either embedded or via a link) from the following sources:		
3.1.3.1.1	• Open Web Application Security Project (OWASP)		
3.1.3.1.2	• Best practices (industry-specific)		
3.1.3.1.3	• Enterprise policies and procedures		
3.1.3.1.4	• Other [Please specify]		
3.1.3.2	Provide examples of "correct" code relevant to the vulnerability detected.		
3.1.3.3	Provide ability to identify a contact or firm with demonstrated subject-matter expert on the problem.		

3.1.4 Hybrid Testing Support

There may be instances where you need to correlate the results of static application security testing (SAST) and DAST testing. If this is the case, the DAST solution shall:		Required	Desired
3.1.4.1	Support the correlation and visualization of SAST and DAST testing results (e.g., results from different types of testing can be integrated into a single dashboard to simplify vulnerability management and remediation).		

3.1.5 Accuracy

The DAST solution shall:		Required	Desired
3.1.5.1	Support techniques to reduce false negatives.		
3.1.5.2	Support techniques to reduce false positives.		

3.1.6 Detection of Security Vulnerabilities

The DAST solution must detect:		Required	Desired
3.1.6.1	Data injection and manipulation to include:		
3.1.6.1.1	• SQL injection		
3.1.6.1.2	• Buffer overflow		
3.1.6.1.3	• Cross-site scripting		
3.1.6.1.4	• Command injection		
3.1.6.1.5	• Cross-site request forgery		
3.1.6.1.6	• LDAP injection		
3.1.6.1.7	• Other [Please specify]		
3.1.6.2	Authentication		
3.1.6.2.1	• Insufficient authentication		
3.1.6.2.2	• Insufficient session expiration		
3.1.6.2.3	• Other [Please specify]		
3.1.6.3.	General needs		
3.1.6.3.1	• Directory indexing and enumeration		
3.1.6.3.2	• File enumeration		
3.1.6.3.3	• Directory and path traversal		
3.1.6.3.4	• Other [Please specify]		
3.1.6.4	Backdoor detection		
3.1.6.5	Other [Please specify]		

3.1.7 Testing Applications Requiring Authentication

The testing of some applications may require the use of authentication credentials associated with the application. The DAST solution must:		Required	Desired
3.1.7.1	Support the following authentication techniques:		
3.1.7.1.1	• Form-based authentication		
3.1.7.1.2	• Browser-based authentication		
3.1.7.1.3	• Automated login		
3.1.7.1.4	• Client-side digital certificates		
3.1.7.1.5	• Other [Please specify]		

3.1.8 Testing Advanced Web Applications and Services

The DAST solution must support the following		Required	Desired
3.1.8.1	Use of client-side code, specifically JavaScript (“single-page applications”)		
3.1.8.2	Web services [If applicable, list specific protocols]		

3.1.9 Testing Interface and Protocols

Applications may use a variety of interfaces and services that are often testing with fuzzing. The DAST solution must provide testing support for:		Required	Desired
3.1.9.1	RESTful-enabled applications		
3.1.9.2	Remote procedure call (RPC)		
3.1.9.3	Session Initiation Protocol (SIP)		
3.1.9.4	Lightweight Directory Access Protocol (LDAP)		
3.1.9.5	Other (Please specify all that your product/service supports)		
3.1.9.6	XML-based protocol fuzzing		
3.1.9.7	Custom (in-house) protocol		

3.2 Analysis and Reporting

3.2.1 Management, Reporting and Analysis

The DAST solution must:		Required	Desired
3.2.1.1	Aggregate results of vendor's DAST tools or tests for further analysis and reporting (e.g., across an enterprise environment where multiple entities may be doing testing in parallel).		
3.2.1.2	Be able to aggregate DAST results with results of third-party application testing tools for further analysis and reporting.		
3.2.1.3	Provide a central dashboard that can consolidate and customize reports based on user role and preferences.		
3.2.1.4	Provide a rating scale for detected vulnerabilities to support:		
3.2.1.4.1	• Standard rating scales		
3.2.1.4.2	• Proprietary rating scale (e.g., based on defining custom vulnerability severities)		
3.2.1.5	Provide a rating mechanism for detected vulnerabilities		
3.2.1.6	Provide standard reports that:		
3.2.1.6.1	• Allow a user (e.g., developer) to quickly focus on the highest severity issues		
3.2.1.6.2	• Can be used to establish regulatory compliance with:		
3.2.1.6.2.1	- PCI		
3.2.1.6.2.2	- HIPAA		
3.2.1.6.2.3	- [Other regulatory requirements]		
3.2.1.7	Allow the development of custom reports		
3.2.1.8	Provide inventory and tracking capabilities to:		
3.2.1.8.1	• Maintain prioritized inventory of externally facing website identified by the solution.		
3.2.1.8.2	• Provide vulnerability inventory to accurately track status of discovered issues over time.		
3.2.1.8.3	• Support management, tracking and reporting status of mitigation of issues identified by the DAST solution.		
3.2.1.9	Support enterprisewide policy management regarding application security (e.g., identify which applications are failing or meeting enterprise policy and standards).		

3.3 Product/Service Environment

3.3.1 Scalability and Performance

The DAST solution shall:		Required	Desired
3.3.3.1	Support [specify number of user seats required with potential growth] number of users.		
3.3.3.2	Allow for the testing of multiple applications simultaneously [specify number with potential growth and/or ask vendor for maximum].		
3.3.3.3	Coordinate multiple testing engines running simultaneously against the same site for scalability.		
3.3.3.4	Provide minimal disruption if run against production applications/systems ("production-safe testing").		

3.3.2 Enterprise Infrastructure Integration

The DAST solution shall:		Required	Desired
3.3.2.1	Integrate with the following [NAME] application development and/or test environments:		
3.3.2.1.1	• Native integration		
3.3.2.1.2	• Via API or plug-in		
3.3.2.1.2	Integrate with the following identity and access management systems for real-time authentication during testing:		
3.3.2.1.2.1	• Active Directory		
3.3.2.1.2.2	• LDAP		
3.3.2.1.2.3	• Oracle Identity Management		
3.3.2.1.2.4	• Other [Please specify]		
3.3.2.1.3	Interface with security and repository infrastructure resources to include:		
3.3.2.1.3.1	• SIEM systems		
3.3.2.1.3.2	• Web application firewalls		
3.3.2.1.3.3	• Internal repository where test results can be queried (e.g., relational database management system based)		
3.3.2.1.3.4	• External repository where test results can be queried (e.g., GitHub)		
3.3.2.1.3.5	• Asset inventory and management tools and databases		
3.3.2.1.4	Allow scheduling of automated tests via:		
3.3.2.1.4.1	• A command-line interface that allows scripting (product)		
3.3.2.1.4.2	• Automatically scheduled requests for testing jobs (SaaS)		
3.3.2.1.5	Support console architecture for analysis and reporting that is:		
3.3.2.1.5.1	• Web-based		
3.3.2.1.5.2	• Thick client (e.g., Win 32 client)		
3.3.2.1.5.3	• Other [Specify]		
3.3.2.1.6	Support stand-alone operation (i.e., tool can be loaded on laptop or desktop and used without further resources such as a central server or access to back-end services).		
3.3.2.1.7	Support internal (behind-the-firewall) testing of internal web applications and monitoring of deployed scanning instances.		

3.3.3 Ease of Use and Productivity			
The DAST solution must:		Required	Desired
3.3.3.1	Support the following user roles:		
3.3.3.1.1	• Application security manager/analyst		
3.3.3.1.2	• Developer		
3.3.3.1.3	• Tester		
3.3.3.1.4	• Penetration tester		
3.3.3.1.5	• Vulnerability manager		
3.3.3.1.6	• Other security professional		
3.3.3.1.7	• Audit professional		
3.3.3.1.8	• Other [Specify]		
3.3.3.2	Be easy to learn and use by end users according to their role.		
3.3.3.3	Provide context-sensitive help.		
3.3.3.4	Provide easily understood system messages.		
3.3.3.5	Require minimum setting of configuration parameters by the end user to establish testing.		

4.0 Support Services

4.0 Support Services	

In order to meet implementation objectives, our organization may require specific services from you for the services listed below. In developing your costs in Section 4.0, please consider the following information: [Tailor this list to meet the needs of "Organization" seeking the DAST solution, including number of users in each category and training delivery method (e.g., online, instructor-led).]

- Training for [insert number of] end users
- Training for [insert number of] system administrators
- Certification program for [insert number of] designated staff
- Implementation services
- Update services
- [Specify level needed for] support services
- Customization [Specify details if known]

4.1	Training: Describe end-user/administrator training courses/options you offer, addressing the following: • Location and method of delivery for training (i.e., live at your location, live on-site, live online, on-demand online) • Mentoring of individual end users by role
4.2	Testing certifications: Provide a list of any security certifications you offer or support that are related to your product. List third parties, if any, that are authorized to administer the certification.
4.3	Implementation: Describe your offerings for supporting the initial configuration and installation of your product/service.
4.3.1	If applicable, provide a list of third parties (e.g., consultants, service providers) that you have certified to provide implementation services on your behalf.
4.4	Program management: Describe your offerings for supporting ongoing program management of DAST assessments.
4.4.1	If applicable, provide a list of third parties (e.g., consultants, service providers) that you have certified to provide program management services on your behalf.

4.5	Maintenance/update services: Describe your offerings, including:
4.5.1	• What services are included in your software maintenance and update program?
4.5.2	• What is your normal revision cycle for standard releases and updates?
4.5.3	• What is the normal distribution path for standard releases? Is it the same path for emergency releases/hot fixes?
4.5.4	• What is the documentation provided with your standard releases? Provide example(s).
4.5.5	• Are ongoing updates to signature packs and attack packs included with standard maintenance, or are they charged separately?
4.6	Support services: Describe your offerings, including:
4.6.1	• What services are included in your support service program? Provide information for both products and services, if you provide both.
4.6.1.1	- Does the DAST service offer vulnerability verification (so false positives are removed before publication)?
4.6.1.2	- Does the DAST service offer login configuration assistance (so the security team can outsource configuration work to the vendor)?
4.6.2	• Is there a knowledge base accessible to end users? To system administrators?
4.6.3	• What are the various levels of standard service that you provide in terms of category of users supported, response time and hours of availability (i.e., Platinum Support means 24/7 two-hour response to all users or offers 24x7x365)?
4.6.4	• Is on-site support available? Provide the terms as outlined in your standard agreement.
4.6.5	• Do you provide consulting services on the process changes necessary to adopt your tools into the software life cycle?

5.0 Pricing and Components

5.0	Pricing and Components

Please provide all solution pricing for the total proposed solution according to the information provided in this RFP.

5.1	Provide a catalog of all items, including hardware, software and support services that are included in your solution(s). This includes a description of each item, whether or not it is optional and its associated list price.
5.2	Describe your pricing/licensing model for enterprise solutions, including any discount tiers. Please indicate any and all limitations to your enterprise pricing.
5.2.1	For each product above, describe how it is licensed (per user, per application, per URL, etc.).
5.2.2	For SaaS solutions, describe how you price the service. Is it by size of application? Is it by contract period? Are there pricing tiers, such as for lightweight, fully automated tests versus more complex tests that require manual intervention by your staff? Do you provide penetration testing, which includes testing outside of the application under test?
5.3	Describe any standard discounting you provide, such as GSA.
5.4	Itemize all items, including hardware, software and support services that you propose for our enterprise, providing a description of each item, its associated list price and its discounted price, if applicable. For each item, indicate which costs are one-time and which items are recurring. Note: If you are providing more than one solution, such as both product and services, list each solution separately according to the instructions above.
5.5	Provide prices for any additional special services, such as on-site end-user training, customization, and certification training (if applicable), according to the information provided in Section 2.0.
5.6	Provide a total cost for each proposed solution, backed by the detail used for developing prices for 5.4 and 5.5.
5.7	Provide a copy of your standard contract together with your typical service level agreement for availability or quality of customer service, or product capabilities.
5.7.1	• What assurances do you provide to the clients that their intellectual property expressed in the code and knowledge of their code vulnerabilities are protected?
5.7.2	• Do you provide a dedicated, named technical account manager?
5.7.3	• What is the process for escalation in case of unsatisfactory or unresolved support issues?

About the Author

Barbara Filkins, a senior SANS analyst who holds the CISSP and SANS GSEC (Gold), GCH (Gold), GSLC (Gold), and GCPM (Silver) certifications, has done extensive work in system procurement, vendor selection and vendor negotiations as a systems engineering and infrastructure design consultant. She is deeply involved with HIPAA security issues in the health and human services industry, with clients ranging from federal agencies (Department of Defense and Department of Veterans Affairs) to municipalities and commercial businesses. Barbara focuses on issues related to automation—privacy, identity theft and exposure to fraud, as well as the legal aspects of enforcing information security in today's mobile and cloud environments.

Sponsor

SANS would like to thank this paper's sponsor:

VERACODE

Last Updated: January 17th, 2019

Upcoming SANS App Sec Training

SANS Las Vegas 2019	Las Vegas, NV	Jan 28, 2019 - Feb 02, 2019	Live Event
Community SANS Silver Spring DEV534 @ SID	Silver Spring, MD	Jan 31, 2019 - Feb 01, 2019	Community SANS
SANS Anaheim 2019	Anaheim, CA	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS Zurich February 2019	Zurich, Switzerland	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Riyadh February 2019	Riyadh, Kingdom Of Saudi Arabia	Feb 23, 2019 - Feb 28, 2019	Live Event
Community SANS Nashville DEV541	Nashville, TN	Feb 26, 2019 - Mar 01, 2019	Community SANS
SANS San Francisco Spring 2019	San Francisco, CA	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS London March 2019	London, United Kingdom	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS Munich March 2019	Munich, Germany	Mar 18, 2019 - Mar 23, 2019	Live Event
Community SANS Denver DEV540	Denver, CO	Mar 25, 2019 - Mar 29, 2019	Community SANS
SANS 2019	Orlando, FL	Apr 01, 2019 - Apr 08, 2019	Live Event
Community SANS Chicago DEV540	Chicago, IL	Apr 15, 2019 - Apr 19, 2019	Community SANS
Cloud Security Summit & Training 2019	San Jose, CA	Apr 29, 2019 - May 06, 2019	Live Event
Community SANS Atlanta DEV540	Atlanta, GA	May 06, 2019 - May 10, 2019	Community SANS
Security West 2019 - DEV522: Defending Web Applications Security Essentials	San Diego, CA	May 09, 2019 - May 14, 2019	vLive
SANS Security West 2019	San Diego, CA	May 09, 2019 - May 16, 2019	Live Event
Community SANS Austin DEV540	Austin, TX	May 20, 2019 - May 24, 2019	Community SANS
Community SANS Vancouver DEV540	Vancouver, BC	Jun 10, 2019 - Jun 14, 2019	Community SANS
SANSFIRE 2019	Washington, DC	Jun 15, 2019 - Jun 22, 2019	Live Event
SANSFIRE 2019 - DEV540: Secure DevOps and Cloud Application Security	Washington, DC	Jun 17, 2019 - Jun 21, 2019	vLive
SANS Cyber Defence Canberra 2019	Canberra, Australia	Jun 24, 2019 - Jul 13, 2019	Live Event
SANS San Francisco Summer 2019	San Francisco, CA	Jul 22, 2019 - Jul 27, 2019	Live Event
SANS Boston Summer 2019	Boston, MA	Jul 29, 2019 - Aug 03, 2019	Live Event
SANS San Jose 2019	San Jose, CA	Aug 12, 2019 - Aug 17, 2019	Live Event
SANS Munich September 2019	Munich, Germany	Sep 02, 2019 - Sep 07, 2019	Live Event
SANS Brussels September 2019	Brussels, Belgium	Sep 02, 2019 - Sep 07, 2019	Live Event
SANS Network Security 2019	Las Vegas, NV	Sep 09, 2019 - Sep 16, 2019	Live Event
SANS Paris September 2019	Paris, France	Sep 16, 2019 - Sep 21, 2019	Live Event
SANS London September 2019	London, United Kingdom	Sep 23, 2019 - Sep 28, 2019	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced