SEARCHSECURITY.COM

technical guide on

APPLICATION SECURITY:

scanning production applications

contents

TechTarget

APPLICATION SCANNING

Scan Strategy

Running security scans against production applications must be carefully planned and focused exercises. BY MICHAEL COBB

IF YOU'RE LUCKY enough to have separate development and test systems, each an exact replica of your live production system, then in theory you don't need to run scans against the production system. But experience shows that even with rigorous change-control processes, identical systems are a rarity; hidden files, different log data, mismatched security configurations and infrastructure differences are common and can have an impact on an application's true security posture. Also enterprise systems with high interconnectivity may be impossible to replicate, while public-facing applications required to maintain PCI-DSS compliance have to be scanned while live.

The main purpose behind scanning an application once it's live is to ensure that the code, operating environment and configuration and security controls put in place to protect it are working as expected and providing the necessary level of confidentiality, integrity, and availability.

The main purpose behind scanning an application once it's live is to ensure that the code, operating environment, and configuration and security controls put in place to protect it are working as expected and providing the necessary level of confidentiality, integrity, and availability. However it can be fraught with problems, and because of this it is often neglected, even slipping off the to-do list completely.

Scans can and will disrupt normal operations unless proper precautions are taken. The main risks are:

- Failure, disruption or reduced performance of the application or connected systems
- Data loss or modification
- Disclosure of data to unauthorized parties
- Test data affecting reports and statistics
- Initiation of irreversible real-world processes
- Triggering of automated responses, countermeasures, and incident handling processes—though testing these may be an objective of the scan
- Reduced awareness of a real incident during the scan
- Violation of legal obligations

Many of these risks can be avoided with careful planning and by focusing your scans on exploring where and how a system is vulnerable to attack and not trying to simulate an actual attack.

WHO SCANS PRODUCTION APPLICATIONS, AND HOW

The head of internal audit should be the person responsible for planning scanning activities and obviously needs to work closely with the heads of IT and operations, both of whom will both need to sign off on the proposed scope of the scan and a suitable timetable. Personnel checks should also be completed on anyone involved in conducting the scan or reporting the results because they may need or obtain access to sensitive information assets.

They should also be independent of the application being scanned so you may have to employ specialist outside consultants unless you have your own dedicated specialized test engineers. (Some regulations require independent third parties to run scans when they are part of an audit.) Using outside consultants can save you the time and expense of choosing and learning how to use an application security scanner.

If you are going to conduct your own scans, make use of the Web Application Security Scanner Evaluation Criteria to select a scanner best suited to your application platform, feature requirements and budget. To be effective, a scanner should allow a combination of manual and automated analysis, be able to utilize information it gathers to form the basis of the checks that follow, and produce easily understood reports. Finally, the scanner should be straightforward to use and offer clear guidelines on what it can check. Access to any software scanners or other tools used for your audits need to be controlled, before, during and after use to prevent any possible misuse or compromise.

To be effective, a scanner should allow a combination of manual and automated analysis, be able to utilize information it gathers to form the basis of the checks that follow, and produce easily understood reports.

You need to define the objectives and scope of any scans to make the most of what will no doubt be a limited test window. Are you're going to scan just the application or the complete environment it operates in and test people, policies, and processes? Or is the scan to check a recent update hasn't adversely affected security?

One objective should certainly be to validate positive security requirements. Positive security requirements are the security controls hopefully documented in the application's business requirements and your security policy. Validating that they function as expected is generally straightforward. For example, "the application will lock out the user after three failed logon attempts" and "all data transmitted from the database to the application front end will be encrypted" can be tested without any real disruption to day to day operations.

But scans also need to test that the application meets negative security requirements, such as not allowing data to be accessed by unauthorized users. By definition, negative

requirements are far more difficult to check but are most likely to be tested and exploited by an attacker. Threat modeling will help you prioritize scans based on the most likely attack scenarios for your particular application.

AVOID DISRUPTION DURING SCANS

These scans have the potential to cause some disruption because the application has to handle abnormal inputs supplied during the scan. Prior to such scans, the team running the scan should know whom to contact and how to contact them in case a problem arises during the scan or an serious vulnerability is discovered that could lead to a total application or system compromise. This contact list should include anyone in the emergency response team (ERT) so that they are aware of the scans taking place. To ensure that everyone is contactable, a timetable for the scans should be agreed in advance.

Assess the possible impact or side effects of any scan beforehand, classifying them low, medium and high. For example, certain multistep HTML forms may be dangerous to automatically fill out with test data, while a slow scan looking for open ports or unauthorized services using a tool such as Nmap is unlikely to cause any problems. Requirements for beginning each category of test can then be drawn up, such as only carrying out high-risk scans out of business hours and with the ERT in attendance, or authenticated scans being restricted to special test accounts.

Assess the possible impact or side effects of any scan beforehand, classifying them low, medium and high.

Obviously the more thorough a scan is, the greater the risk of disruption. This could be anything from flooding customer support with error emails or creating a denial of service (DoS) condition due to open connections, bandwidth and memory usage by the scanner. Scans should be interruptible at any point if unintended consequences are observed or reported, particularly those that produce a large number of test inputs, such as fuzzing, or aim for DoS conditions as their primary effect.

Good scanning tools support throttling so if system performance degrades for any reason, the scan automatically slows down or stops until normal performance returns before resuming. If possible, scans should only collect enough evidence to conclude that a vulnerability is likely instead of testing for the actual defect. An obvious example is scanning for known vulnerabilities in software packages by checking their version and patch levels instead of trying to exploit the actual vulnerability.

If it's necessary to demonstrate that a vulnerability can be exploited, then the payload should show a visible or measurable effect without any undesirable side effects. The WhiteHat Sentinel scanner for example uses proprietary pseudo-code. This enables the scanner to identify vulnerabilities without the payload being interpreted by parsers within the application.

Wherever possible the root cause of any problems that are discovered should be investigated. A SQL injection vulnerability, for example, may be due to inadequate

input validation or a coding error that means data doesn't get passed to the validation function. This is obviously a lot easier if the source code is available because static analysis can determine the likelihood of the vulnerability being exploited and its risk rating. All suspected vulnerabilities should be recorded and ranked in terms of seriousness to enable the business owner to prioritize vulnerabilities for remediation.

Removing data created by any scans is an important post-scan task. Business users don't want to receive inflated or erroneous figures, such as page hits or inquiries, caused by a high volume of test data. Where possible, scanners should use inputs that are easy to identify and should always record the user accounts and IP addresses used in the scan along with the start and end times of each scan. These precautions will also help avoid overreaction if the scanning is detected by operations personnel and interpreted as an attack.

THOROUGH REPORTING VITAL TO IT AND BUSINESS OWNERS

After the completion of any scans, a formal report should be produced covering what scans were performed, by whom, when, and details the findings. It must make clear where any vulnerabilities or concerns exist together with an assessment of their impact and any recommended mitigation strategy. A good report will provide an executive summary giving a picture of the overall state of the security of the application and what needs to be done to improve it.

A good report will provide an executive summary giving a picture of the overall state of the security of the application and what needs to be done to improve it.

If the test is a follow-up to check that earlier recommendations have been correctly implemented, or part of an ongoing testing program, then provide a comparison to previous tests. This comparison must highlight any persistent areas of weakness such as vendor patches not being applied by system administrators, or systems being incorrectly configured.

These reports are of interest not just to the IT department, software development team and auditors, but to key stakeholders such as system and business owners so they should explain in layman's terms the more complex technical issues. You may want to consider issuing two versions, one containing detailed analysis of the vulnerability for the IT and development teams, and one without for other stakeholders. They will contain sensitive information so the report should be classified and distributed only to authorized individuals.

Any limitation that you impose on the set of scans allowed and the time to complete them will of course result in an incomplete and inaccurate picture of how secure the application actually is. This doesn't mean that they're not worth carrying out, but given that there are limits to what can be tested on a production system, actively monitoring it is vital to provide defense in depth.

LIVE SCANS TAKE YOU BEYOND LOG MONITORING

Application and system monitoring provides details of what is happening and what has happened. It provides security against lapses in your perimeter and application defenses by alerting you to problems so that defensive measures can be taken potentially before any real damage is done. Without monitoring a live application, you have little chance of discovering whether it has been compromised.

Simply purchasing and deploying a log management solution won't provide you with any additional security. You have to use the information collected and analyze it on a regular basis; for a high-risk application, this could mean automated reviews on an hourly basis. To be effective, logs need to have accurate timestamps and to record as a minimum user activity and any traffic crossing a network boundary. Monitoring is far less obtrusive and disruptive than scanning but it can take time to investigate unusual events.

Any suspicious behavior or critical events must generate an alert that is assessed, categorized and acted upon in accordance with your policy on responding to information security events. However your log files are worthless if you cannot trust their integrity.

The first thing most hackers do is try and alter the log files to hide their presence.

The first thing most hackers do is try and alter the log files to hide their presence. To protect against this, you should record logs locally and to a remote log server. This provides redundancy and an extra layer of security because you can compare the two sets of logs—any differences indicative of suspicious activity. If you can't stretch to a dedicated log server, logs should be written to a write-once media to prevent an attacker overwriting them to try and avoid detection.

Scanning and monitoring live applications plays an important role in keeping them secure, allowing you to fix problems before a potential attacker can take advantage. Scheduled scanning helps ensure security is maintained over time, particularly as new threats emerge and changes to system configurations alter the application environment. But just because a scan doesn't find any problems doesn't mean that they don't exist. This is why it needs to be combined with active monitoring in order to stop anyone who's trying to exploit a vulnerability you didn't find.

Michael Cobb CISSP-ISSAP, CLAS, is a renowned security author with more than 15 years of experience in the IT industry. He is the founder and managing director of Cobweb Applications, a consultancy that provides data security services delivering ISO 27001 solutions. He co-authored the book IIS Security and has written numerous technical articles for leading IT publications. He is the guest instructor for several SearchSecurity.com Security Schools and, as a SearchSecurity.com site expert.

TECHTARGET SECURITY MEDIA GROUP

SearchSecurity.com

EDITORIAL DIRECTOR Michael S. Mimoso

SEARCHSECURITY.COM
SENIOR SITE EDITOR Eric Parizo

NEWS DIRECTOR Robert Westervelt

SITE EDITOR Jane Wright

ASSISTANT EDITOR Maggie Sullivan

ASSOCIATE EDITOR Carolyn Gibney

ASSISTANT EDITOR Greg Smith

ART & DESIGN
CREATIVE DIRECTOR Maureen Joyce

VICE PRESIDENT/GROUP PUBLISHER
Doug Olender

PUBLISHER Josh Garland

DIRECTOR OF PRODUCT MANAGEMENT
Susan Shaver

DIRECTOR OF MARKETING Nick Dowd

SALES DIRECTOR Tom Click

CIRCULATION MANAGER Kate Sullivan

PROJECT MANAGER Elizabeth Lareau

PRODUCT MANAGEMENT & MARKETING
Corey Strader, Andrew McHugh,
Karina Rousseau

SALES REPRESENTATIVES
Eric Belcher ebelcher@techtarget.com

Patrick Eichmann
peichmann@techtarget.com

Leah Paikin lpaikin@techtarget.com

Jeff Tonello jtonello@techtarget.com

Nikki Wise nwise@techtarget.com

TECHTARGET INC.
CHIEF EXECUTIVE OFFICER Greg Strakosch

PRESIDENT Don Hawk

EXECUTIVE VICE PRESIDENT Kevin Beam

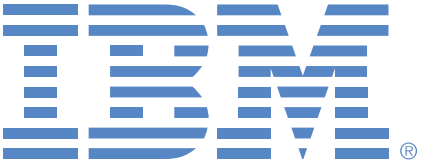
CHIEF FINANCIAL OFFICER Jeff Wakely

EUROPEAN DISTRIBUTION
Parkway Gordon Phone 44-1491-875-386
www.parkway.co.uk

LIST RENTAL SERVICES
Julie Brown
Phone 781-657-1336 Fax 781-657-1100

TechTarget
Where Serious Technology Buyers Decide

"Technical Guide on Application Security" is published by TechTarget, 275 Grove Street, Newton, MA 02466 U.S.A.; Toll-Free 888-274-4111; Phone 617-431-9200; Fax 617-431-9201.



- **Securing today's applications: Design, deliver and secure smarter software and services**
- **It's hard to tell the good guys from the bad, make sure you are one of the good guys**
- **Realize cost reduction opportunities with automated application security testing**

About IBM:
At IBM, we strive to lead in the creation, development and manufacture of the industry's most advanced information technologies, including computer systems, software, networking systems, storage devices and microelectronics. We translate these advanced technologies into value for our customers through our professional solutions and services businesses worldwide.