Key Technology Research on Backtracking Attack Eventof Government Website Comprehensive Protection System

Chen Chen[a,*], Rui Wang[b], Haiwei Li[c], Yijun Wang[d]

Information Security Department, First Research Institute of the Ministry of Public Security of PRC, Beijing China

[a]hazel_cc@163.com, [b]ruiwang114@163.com, [c]lhwivan@163.com, [d]w-y-j-tc@163.com

*corresponding author

Keywords: Interactive Application Security Test technology; Runtime application self-protection technology; kernel reinforcement technology; log analysis; backtracking attack event

Abstract: There are more and more attack that threaten the security of system application in the complex network environment. In this paper, a government website comprehensive protection system (G01) is proposed for government departments and important enterprise users to understand the weakness of their application systems. The government website comprehensive protection system (G01) is aimed at making timely and effective response against attacks from the network. Event description method in G01 is different from the previous alarm based attack log methods, which makes the attack more intuitive, clear and readable. Key technology points of backtracking attack event are detailed introduced, including the key technology of log acquisition, such as Interactive Application Security Test, Runtime Application Self-Protection, Kernel Reinforcement and Attack log analysis. G01 has high accuracy rate because it can collect logs at every step of an attack, in order to collect logs together and form security events. Attack log analysis algorithm adopted by this system determines the accuracy of the final event presentation.

1. Introduction

With the development of the global Internet and e-commerce, Amount of website attack log has sharply increased. Daily log volume of hundreds of millions or billions has become a common situation. Extraction relationship is one of difficult problems faced by current website attack log analysis. With the continuous changes and development of network attacks, website application systems use plenty of protection methods to defend network attacks, including website application-level intrusion prevention systems, cloud security defense systems, server system firewalls, and server system anti-virus software. However, logs generated by various devices are difficult to relate, and causing another problem of log analysis. With the division of labor in the whole society, each unit runs its own application system within its business scope. Most units only defend and analyze the network attacks generated by their own application systems while do not form a whole attack situation for warning and analysis, and also limite log analysis.

Application systems of government departments and important enterprises record various attack log information as large amount of logs, multiple types of logs, and scattered information, etc. For just one application, amout of differect types of log files are extremely large, and saved with various formats of storage. It is difficult to describe and relate attack events and threats. Moreover, it is difficult to understand the attack situation of other application systems and form a comprehensive evaluation and analysis of the network attack.

Therefore, it is an important challenge to analyze mass and complex security logs into readable threat events, and identifing high-risk vulnerabilities. It is also an challenge to link attack events in order to help cyber security regulators assessing the overall attack situation and threat level.

Based situations above, we designed and implemented a Government Website Comprehensive Protection System (G01). Through deploying probes at nodes of web application systems, G01 could collect logs generated by each step of the attack in unified format, and extract and correlate related

information from mass logs to discover attack events, automatically tracing the attack process, and intuitively shows the intercepted attacks and possible hazards. By this schema, G01 could help the government and enterprises to quickly locate and repair the risk points used by hackers.

2. Key technologies of attack event backtracking

2.1 Technical architecture of attack event backtracking

Firstly, G01 captures traces of each step of network attack from network layer, system layer, and kernel layer. Then, recording the logs, extracts and correlates an attack base on attack log analysis algorithm. Finally, traceing out the complete attack event. Detailed process is shown as Figure 1.

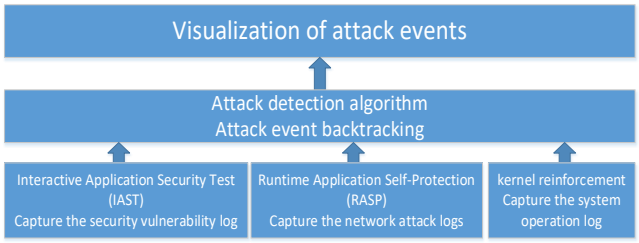

Figure 1 Attack event backtracking technical architecture diagram of G01

G01 cananalye logs and extract events with high efficiency amd accuracy, for the system captures logs at every step of the attack and deploys attack detection algorithm that determines the accuracy of the final event presentation.

Interactive Application Security Test (IAST) is used to capture security vulnerability logs, Runtime Application Self-Protection (RASP) is used to capture network attack logs, and kernel reinforcement is adopted to capture operating system logs. Network attack logs and system operation logs are calculated to analyze and extract events, and security vulnerability logs are used to verify attack events. G01 apparently reduce the false positive rate andfalse negative rate, and analyzing the complete attack event path, hazard degree and protective measures.

2.2 Key points of log capture

- Interactive Application Security Test (IAST) for security vulnerability logs capture

IAST is a new vulnerability analysis technology first proposed by Synopsys. It combines Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) which aims to provide increased accuracy of application security testing through the interaction of the SAST and DAST. IAST provides the best solution for SAST and DAST. This approach can confirm or disprove the exploitability of the detected vulnerability and determine its point of origin in the application code. IAST executes in the application, and can continuously monitors and identifies vulnerabilities.

IAST works by hooking into the application, and making internal analysis as the application runs. Just as a debugger would do, IAST looks into code execution in memory and seeks out specific events such as database queries, file system access, web service calls, input validations, etc., aims to discover how these events cause vulnerabilities.

G01 uses IAST technology to collect and monitor runtime function execution, data transmission, and real-time interaction in web application by deploying an agent on the server side. Server-side probes interact with the console scanner in real time. By this way, G01 efficiently and accurately identifies security defects and vulnerabilities, locates code files, lines, functions, and parameters of vulnerability. A basic implementation process is as follows:

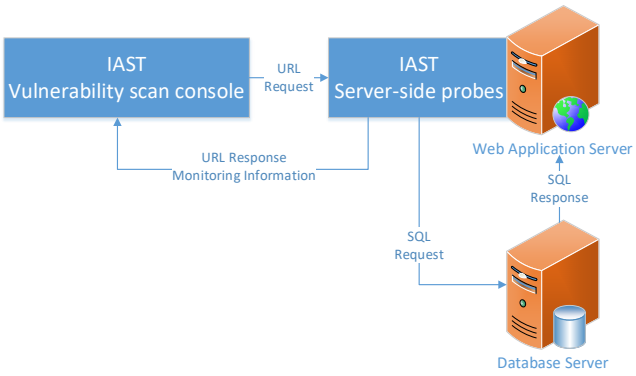

Figure 2 Schematic diagram of IAST used by G01

G01 also uses IAST to detect and record system application vulnerabilities in real-time and dynamic interactions to form a security vulnerability log.

- Runtime Application Self Protection for network attack logs capture

Runtime application self-protection (RASP) is built or linked into an application or application runtime environment, and is capable of controlling application execution, detecting and preventing real-time attacks.

RASP security products integrate with an application to prevent attacks at runtime by monitoring and analyzing traffic and user behaviors. When detecting an attack, RASP products make alerts, block application execution for individual requests, and sometimes virtually patch the application to prevent further attack. They typically integrate with an application at either the language runtime or application server layer, which gives them function-level code visibility into the application. This visibility allows them to identify attacks more accurately, reducing false positives, and reporting or blocking only those actions which constitute legitimate threats.

G01 use deep analysis of Runtime application self-protection (RASP) to block potentially malicious behavior without learning cost which Web Application Firewall (WAF) needs, and with potentially greater accuracy. G01 also use RASP's deep analysis to understand common vulnerabilities and attack techniques and adjust their policies, technical controls, and other mitigation efforts accordingly.

G01 embeds the RASP security protection code into the running server application. When a suspicious request connects the application system, G01 does not intercept it immediately but marks it firstly, then checks whether the output responding is dangerous, finally achieving application self-protection and minimizing the probability of false positives and false negatives. Detail process of RASP is shown in Figure 3:

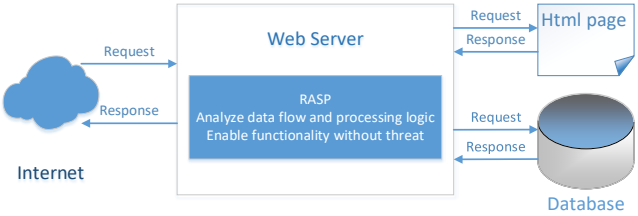

Figure 3Schematic diagram of Runtime application self-protection technology used by G01

G01 uses RASP to continuously monitor the network traffic, context and behavior of the application system, identifying and defending against known and unknown threats. By this way, G01 can effectively record and defend against SQL injection, command execution, file upload, arbitrary file read/write , deserialization, struts2, and other application vulnerabilities which could not effectively protected based on traditional signature methods. G01 uses RASP and IAST in combination to improve accuracy and reduce misjudgment.

- Kernel reinforcement for system operation logs capture

Kernel Reinforcement (KR) technology sets an autonomously controllable security shell at the kernel of the operating system to intercept the access behavior of threats to system resources, in the

purpose of preventing certain software from modifying and deleting system files, and protecting operating system security.

G01 strengthens the operation system through kernel hooks, and improves the security and anti-attack ability of the operation system. G01 uses this technology to protect system files, prohibiting malicious modification of files, prohibiting malicious code execution, and prohibiting loading of drivers without digital signatures. G01 uses this technology to stop useless services in the operating system, therefore reducing system resource usage and improving system security. G01 uses this technology to provide a comprehensive and efficient baseline check on the operating system security configuration, such as system weak passwords, checking cloned accounts, scheduled tasks, etc. as shown in Figure 4.

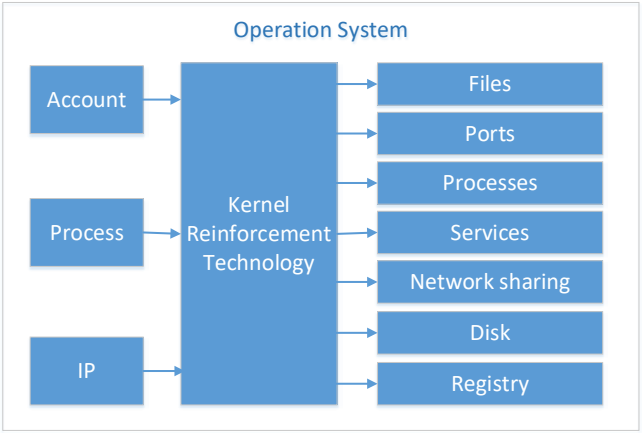

Figure 4 Schematic diagram of the kernel reinforcement technology used by G01

G01 uses KR to protect important files and functions of the operating system, and also uses KR to record logs from user layer, service layer, and driver layer in operating system.

2.3 Attack log analysis

- Attack event backtracking

G01 adopts a range of advanced protection technologies, e.g. web intrusion detection algorithm, web authority upgraded detection algorithm and operating system intrusion detection algorithm to analyze the logs, finally describing the whole process of threat attack events as shown in Figure 5.

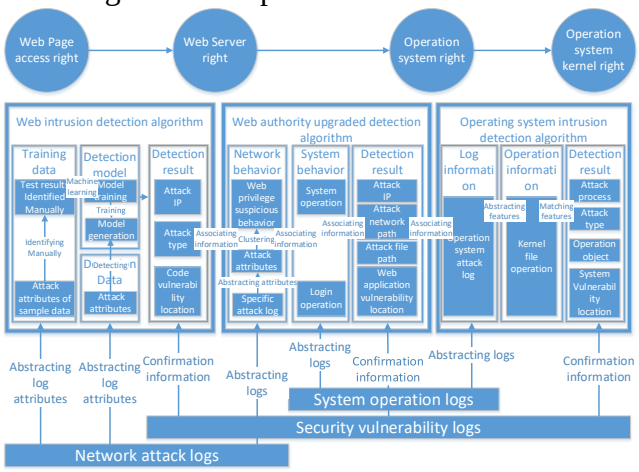

Figure 5 Algorithm diagram of attack event backtracking by G01

- Web intrusion detection algorithm

Abstracting log attributes: G01 abstracts attack attributes from the network attack logs, e.g. web access frequency, web access time interval, time, geographic location, time zone, return byte length, attribute character distribution, occurrence and absence of attributes, attribute character frequency, path, call and access order, etc.), applying classification algorithms, e.g. Bayesian, decision tree, SVM, etc.) for classification.

User identification and session identification: Different IP addresses are treated as different users, and client with different IP addresses are treated as different users. The session is recognized by 30 minutes.

Learning phase: the algorithm of mining association rules is used to calculate the frequency of attribute occurrence. In the learning phase, the normal behavior model is established. Once the normal behavior model is established, the system will switch to the detection phase.

Detection phase: After switching from the learning phase, it is compared with normal behavior, and if there were abnormal behaviors, it is considered to be an invasion. If finds sensitive keywords such as select, insert, update, <script>, exec, etc. were discovered, this request would be considered as SQL injection or CSS attack.

Confirmation phase: use the security vulnerability logs to confirm the detected attack corresponding to the web application code location.

Expected effect: Web intrusion detection algorithms are used to analyses attacks of web application layer, output the attacker IP addresses, web application attack types, code vulnerability locations.

- Web authority upgraded detection

Abstracting logs: In the network attack logs, the search meets the conditions {User-Agent: short content, old version; Referrer: empty; URIs: first appearance; client behavior characteristics: only access one URI, the same User-Agent uses multiple IP addresses within one day}, called a collection of specific attack log results.

Abstracting attributes: The algorithm abstracts attack attributes in the specific attack logs(such as access frequency, access time interval, time, geographic location, time zone, return byte length, attribute character distribution, occurrence and absence of attributes, attribute character frequency, path, call or Access order, etc.). Using several attributes as variables of multidimensional vectors, the algorithm applying a clustering algorithm to find web privilege suspicious behavior.

System behavior: In the system operation logs, the algorithm finds the operations of privileged users, including login operations and system operations, such as creating users, restarting processes, clearing disk space, reading logs, etc.), and if there is an operation, give the operation time.

Abstracting results: Administrator login records before and after the above time are searched in the system operation logs. If there is no administrator login record in this time range, the algorithm searches for access logs within this time range in the network attack logs, abstracting URI from the logs as output of the algorithm.

Confirmation phase: The security vulnerability log is used to confirm the detected attack corresponding to the web application, web server and web application vulnerability location.

Expected effect: The algorithm gives the attack path, including network path and file path, that is, identify the file as WEBSHELL. The algorithm also outputs the IP address or UID of the suspected attacker and the vulnerability location of web application.

- Operating system intrusion detection

Abstracting logs: The attack session log is abstracted from the system operation logs. Log attributes include the operation subject (process, user, IP) and operation object (file, port, process, service, network share, disk, registry).

User login and logout: the user login operation attributes (such as login number, login frequency, login time, login interval, login user name, login user password, login success, etc.) are detected from the system operation logs. Feature matching algorithm is used to detect specific attack types.

Confirmation phase: Use the security vulnerability log to confirm that the detected attack corresponds to the location of the operation system vulnerability.

Expected effect: The algorithm gives the process used by the attacker, the files accessed by the operation system (operation object), the attack type and the location of the operating system vulnerability.

2.4 Visualization of attack events

- Multi-dimensional visualization

G01 exhibits attack events multi-dimensionally, including the list of attack events, the directed graph of attack process and the detailed progress of attack.

The attack event lists describe main information of each attack event, including the event summary list (e.g. attack time, attacked server, event type, event grouping, attack source IP, target IP, attack phase, risk level, status, etc.), event descriptions, and processing comments.

Directed graph of attack process shows the attack source IP addresses for network attacks and system attacks. As shown in Figure 6, the detailed process of an attacker penetrates from web page into the web system, till deep into the operating system, and has displayed key node commands and important information.

The progress of an attack shows each step of the attack with time clues, including attack commands, attack descriptions, attack results, etc., as shown in Figure 6:

Figure 6 Detailed process diagram of the attack event by G01

3. Conclusion and Prospect

The government website comprehensive protection system (G01) using security event in stand of alarm based attack log to describe process of an attack, makes the attack more intuitive, clear and readable. It is of revolutionary significance to the protection and repair of government websites and enterprise websites.

Based on the huge number of organized security events, the G01 analyzes the trend and the degree of damage of all installed servers, which reflect the attack on important infrastructure. Protecting is G01 permanent direction.

References

[1] Gartner Identifies the Top 10 Technologies for Information Security in 2014

https://www.gartner.com/newsroom/id/2778417

[2] Interactive Application Security Testing

http://www.quotium.com/resources/interactive-application-security-testing

[3] The What, Why and Who of Runtime Application Self-Protection (RASP)

https://www.synopsys.com/blogs/software-security/runtime-application-self-protection-rasp/

[4] GUO Xiaolei Research on Web Users Clustering Based on Web Log Mining. Chinese Master's Theses Full-text Database, 2009.2

[5] CHU Weiming, HUANG Jin, LIU Zhile. Research on Collecting Data for Situation Awareness of Cyber Space[J]. Netinfo Security, 2016(9):202-207.