|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | KITRI 모의해킹 28기 | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  |  | | |  | |
|  | Insufficient Logging  And Monitoring | | | | | | |  |
|  | | 불충분한 로깅 및 모니터링 | | | |  | | |
|  | |  | | | |  | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |

|  |
| --- |
| 1. 개요 |
| 참조 : <https://owasp.org/www-project-top-ten/2017/A10_2017-Insufficient_Logging%2526Monitoring> |
| 웹 애플리케이션의 불충분한 로깅과 모니터링은 침해 사고 탐지 및 대응의 지연 및 누락과 함께 공격자들이 시스템을 더 공격하고, 지속적으로 더 많은 시스템을 공격할 수 있도록 만들고, 공격자들이 데이터를 변조, 추출 또는 삭제할 수 있게 된다. |
| 발생하는 상황 |
| * 로그인, 로그인 실패, 고액 트랜잭션 등 검증 가능한 이벤트는 기록되지 않는다. * 경고와 오류는 로그 메시지를 불충분하거나 불명확하게 만든다. 여기에는 법의학자가 이해할 수 있는 충분한 세부 정보가 없는 모호한 오류 로깅이 포함된다. * 의심스러운 활동에 대해서는 응용 프로그램 및 API 로그를 모니터링하지 않는다. * 로그는 로컬에서만 저장된다. 백업되지 않은 로그는 시스템에 액세스하는 침입자에 의해 삭제될 위험이 있다. 이렇게 하여 침입자는 자신의 흔적을 감추어 침입의 근원을 추적할 수 없게 된다. * 적절한 경보 임계값 및 대응 에스컬레이션 프로세스가 없거나 비효율적이다. * DAST 도구에 의한 침투 시험 및 스캔(예: OWASP ZAP)은 어떤 경고도 트리거하지 않는다. * 응용 프로그램은 실시간으로 활성 공격을 탐지, 에스컬레이션 또는 경고할 수 없다. |
| 대응 방법(OWASP 권고사항) |
| * 모든 로그인, 액세스 제어 및 서버측 입력 유효성 검사 오류는 의심스러운 계정이나 악의적인 계정을 식별하기에 충분한 사용자 컨텍스트로 기록되어야 한다. 로그는 법의학적 분석이 지연될 수 있는 기간 동안 보관되어야 한다. * 중앙 로그 관리 도구에서 쉽게 사용할 수 있는 형식으로 로그를 만드십시오. * 고액 거래는 조작이나 삭제를 방지하기 위해 무결성 통제가 있는 감사 추적이 있어야 한다. * 의심스러운 활동을 적시에 탐지하고 대응할 수 있도록 효과적인 모니터링 및 경보를 확립해야 한다. * *NIST 800–61 rev. 2*와 같은 사고에 대한 대응 및 복구 계획이 수립되어야 한다. |
| 추가 보안조치(IBM) |
| * 감사 로그에 이벤트를 캡처하고 저장하기 위한 별도의 보안 강화 서버 플랫폼. * 시스템 시계를 동기화하기 위한 네트워크 시간 동기화 기술의 사용. 이를 통해 자동화된 모니터링 툴이 실시간으로 발생하는 이벤트 패턴을 분석할 수 있다. * 로그에 대한 강력한 액세스 제어 * 공식적인 사고 대응 계획 작성. * 모니터링 담당자에 대한 경고 시스템 구현을 통한 상시 모니터링 보장 |

|  |
| --- |
| 1. Web Application Firewall |
| **네트워크 방화벽의 동작원리는 네트워크에 돌아다니고 있는 패킷이 정상인지 아닌지를 판단한다.**  **반면에 웹 방화벽은 패킷이 아닌 사용자의 URL이 정상이고 올바른 요청인지 판단을 하게된다.** **OSI 7 Layer에 비교를 해보도록 하자.**  **일반 방화벽은 패킷이 돌아다니게 되는 L3~L4(Network Layer)레벨에 동작한다.**  **웹 방화벽은 HTTP/HTTPS 요청이 있는 L7(Application Layer)레벨에서 동작한다.** |
| WAF( Web Application Firewall) |
| **웹 사이트의 구조와 동작원리는 아래 사진과 같다.**    **웹 공격의 대부분은 Web Application을 구축할때 생겨나는 취약점을 이용해서 웹서버를 공격하거나 DB의 내용을 악용하여 정보를 탈취하거나 노출을 시킨다.** **공격자는 HTTP Request에 특정 공격코드 또는 특정 Web Application만이 가지고 있는 취약점을 우회하는 코드를 삽입하여 웹 서버에 전송하게 되는데 여기서 Web Application은 의도하지 않은 동작을 하게 되는데 그 결과를 HTTP Reply를 통해 공격자에게 다양한 정보들을 전송하게 된다.**  **이때 필요한 것이 웹 방화벽이다.** **웹 방화벽이란 웹 서버쪽으로 전송되는 모든 HTTP Request Packet을 검사하여 Web Application에게 의도하지 않은 내용들을 전송하지 못하도록 막는 역할을 하는게 웹 방화벽이다.** **또한 웹 서버에서 통과하는 HTTP Reply Packet 내용을 감시하여 특정 정보의 유출을 막는 역할도 한다.**  **Proxy Server의 원리를 가져 온 것이다** **Proxy Server는 Client와 Server간의 통신을 중계하고 Relay하는 역할을 한다.** **결론적으로 웹 방화벽의 원리는 웹 서버에 들어오고 나가는 모든 패킷을 Proxy Serve원리를 적용하여 패킷을 내용을 검사하고 차단하는 것이다.** |

|  |  |
| --- | --- |
| 1. 웹 애플리케이션 로그관리 | |
| 참고 : <https://www.kisa.or.kr/notice/press_View.jsp?mode=view&p_No=8&b_No=8&d_No=1753>  KISA에서 권고하는 로그설정 노트(응용프로그램)를 참고하였다.  위 문서는 logrotate를 사용하여 로그를 순환관리하고 있다.  \* **logrotate란?** 로그 파일(시스템 로그)을 rotates, compresses, and mails 을 할 수 있다.  설정 파일을 변경해도 관련 프로세스를 새로 시작할 필요 없이 cron 데몬이 주기적으로 실행 시켜준다. | |
| apache 로그 설정 | |
| 1. /etc/logrotate.d/httpd 생성  # touch /etc/logrotate.d/httpd  2. /etc/logrotate.d/httpd 설정 (아래 항목을 작성)  /var/log/httpd/\*log {  daily  create 0664 root root  missingok  notifempty  sharedscripts  rotate 180  postrotate  /sbin/service httpd reload > /var/log/httpd/rotate 2>&1 || true  endscript  compress  } | 로그 파일은 파일별 하루의 로그 저장  로그 파일 생성 시, 0664의 권한, root 소유, root 그룹  로그 파일이 존재하지 않아도 오류 발생하지 않음  로그 파일은 6개월 분량 보관  로그 파일명에 YYYYMMDD 형식의 문자열을 추가  로그 내용이 없으면 rotation 하지 않음  로그파일이 로테이트된 후 다음의 커맨드를 실행  /sbin/service http reload > /var/log/httpd/ rotate 2>&1 || true |
| Nginx 로그 설정 | |
| 1. /etc/logrotate.d/nginx 생성  # touch /etc/logrotate.d/nginx  2. /etc/logrotate.d/nginx 설정 (아래 항목을 작성)  /var/log/nginx/\*log {  create 0644 nginx nginx  daily  rotate 180  missingok  notifempty  compress  sharedscripts  postrotate  /bin/kill -USR1 `cat /run/nginx.pid 2> /dev/null` 2>/dev/null || true  Endscrip  } | 로그 파일은 파일별 하루의 로그 저장  로그 파일 생성 시, 0664의 권한, root 소유, root 그룹  로그 파일은 6개월 분량 보관  로그 파일이 존재하지 않아도 오류 발생하지 않음  로그 파일명에 YYYYMMDD 형식의 문자열을 추가  로그 내용이 없으면 rotation 하지 않음  로그파일이 로테이트된 후 다음의 커맨드를 실행  /bin/kill -USR1 `cat /run/nginx.pid 2>/ dev/null` 2>/dev/null || true |
| IIS 웹 서버 로그 설정 변경 | |
| 1. 로그 경로 변경 (추가 볼륨 사용 권장)  IIS 관리자(또는 inetmgr.exe 실행) ▶ 로깅 :  로그 저장 옵션은 디폴트(매일 다른 파일로 로그 저장) 유지 권장  다음과 같은 추가 볼륨 경로로 변경  : <추가볼륨>\IIS\_log  웹 로그는 추가 disk 확보 후 최소 6개월 이상의 분량을 저장 권고 (보유기간 설정은 따로없어 스크립트를만들어 작업 스케쥴러를 사용하여 정기적으로 삭제관리한다)  참고 : <https://docs.microsoft.com/en-us/iis/manage/provisioning-and-managing-iis/managing-iis-log-file-storage> | |

|  |
| --- |
| 3. ModeSecurity |
| ModSecurity란  아파치 웹 서버에서 동작하는 오픈 소스 웹 방화벽으로 웹 공격에 대한 침입방지 기능을 추가해 주는 모듈  \*Nginx에도 ModeSecurity를 적용 할 수 있다, 다만 수동으로 설정해주어야 하는 부분이 많다. |
|  |
| **OWASP Rule 적용** |
| **modsecurity 탐지 테스트**    \* 'var/log/apache2/error.log' 에서 탐지로그를 확인할 수 있다 |

|  |
| --- |
| 4. 웹 애플리케이션에서의 처리 |
| **혹은 웹 애플리케이션의 코드 로직에서 의심스러운 항목을 로그로 남기도록 처리가 가능하다.** |
| **다중 로그인 시도** |
| **function login($userName,$password){**  **if(authenticate($userName,$password)){**  **return True;**  **}else{**  **incrementLoginAttempts($userName); if(recentLoginAttempts($userName) > 5){**  **writeLog("Failed login attempt by User: " . $userName . " at " + date('r') );**  **}**  **}**  **위 코드는 5번이상 로그인 실패 시 시간과 유저이름을 로그에 남기는 코드이다, 이러한 방식으로 소스코드 내의 추가하여 남기는 것도 방법이 될 수 있다.** |