http://dx.doi.org/10.3745/KTCCS.2013.2.9.405

Security Check Scheduling for Detecting Malicious Web Sites

Choi Jae Yeong[†] · Kim Sung Ki[††] · Min Byoung Joon[†††]

ABSTRACT

Current web has evolved to a mashed-up format according to the change of the implementation and usage patterns. Web services and user experiences have improved, however, security threats are also increased as the web contents that are not yet verified combine together. To mitigate the threats incurred as an adverse effect of the web development, we need to check security on the combined web contents. In this paper, we propose a scheduling method to detect malicious web pages not only inside but also outside through extended links for secure operation of a web site. The scheduling method considers several aspects of each page including connection popularity, suspiciousness, and check elapse time to make a decision on the order for security check on numerous web pages connected with links. We verified the effectiveness of the security check complying with the scheduling method that uses the priority given to each page.

Keywords : Security Check, Detection Scheduling, Malicious Web Page, Web Crawling

악성사이트 검출을 위한 안전진단 스케줄링

최 재 영[†] · 김 성 기[††] · 민 병 준[†††]

요 약

최근의 웹은 구현 방법과 이용 패턴이 변화되면서 서로 연결되고 융합되는 형태로 변화하였다. 서비스가 진화되고 사용자 경험이 향상되었으나 다양한 출처의 검증되지 않은 웹자원들이 서로 결합되어 보안 위협이 가중되었다. 이에 웹 확장의 역기능을 억제하고 안전한 웹서비스를 제공하기 위해 확장된 대상에 대한 안전성 진단이 필요하다. 본 논문에서는 웹사이트의 안전한 운영을 위해 안전진단을 외부 링크까지 확장하여, 진단 대상을 선별하고 지속적으로 진단하여 악성페이지를 탐지하고 웹사이트의 안전성을 확보하기 위한 스케줄링 방안을 제안한다. 진단 대상의 접속 인기도, 악성사이트 의심도, 검사 노후도 등의 특징을 추출하고 이를 통해 진단 순서를 도출하여 순서에 따라 웹페이지를 수집하여 진단한다. 실험을 통해 순차적으로 반복 진단하는 것보다 순위에 따라 진단 주기를 조정하는 것이 중요도에 따라 악성페이지 탐지에 효과적임을 확인하였다.

키워드 : 안전진단, 탐지 스케줄링, 악성웹페이지, 웹크롤링

1. 서 론

최근의 웹사이트는 하이퍼링크에 의한 단순한 연결을 넘어 매쉬업에 의한 연계로 서비스가 융합되며 새로운 가치를 창출하고 있다. 이를 지원하기 위해 기술적으로 웹사이트를 구성하는 요소들의 결합이 느슨해지고 다양한 출처의 검증되지 않은 컨텐츠 유입이 증가하였다. 이에 따라 해커의 공격도 다양해지고 파급 효과도 커져갔다. 공격 대상 서버에 집중하던 공격 패턴은 연계 서버와 클라이언트까지 공격 대상으로 삼았다. 이에 따라 웹보안은 내부의 문제로 국한되지 않는다. 연결된 지점에 대한 보안 강화가 요구되며 지속적인 감시가 필요하다. 그러나 한정된 자원으로 급속히 성장하고 변화하는 분산된 원격지의 모든 웹사이트의 안전성을 진단하는 것은 한계가 있다. 진단 대상의 변경을 알 수 없기 때문에 지속적으로 감시하더라도 진단 간격이 발생한다. 또한 대규모의 웹사이트를 짧은 간격으로 진단하려면 많은 비용이 소요된다.

그러므로 대상을 선별하고 대상의 특성에 따라 진단 간격을 차등하게 부여하여 진단하는 방안이 필요하다. 진단 대상을 차별화하여 주요 대상은 보다 짧은 주기로 자주 진단하고 상대적으로 중요도가 떨어지는 이용이 적은 대상은 우선 순위를 낮추어 긴 주기로 진단하는 것이 효율적이다. 이를 위해 본 논문에서는 진단 대상 선정과 가중치 부여 방식을 제시하고 지속적으로 진단하기 위한 스케줄링 방안을 제안한다.

※ 이 논문은 인천대학교 자체연구비 지원에 의하여 연구되었음.
† 준 회 원 : 인천대학교 컴퓨터공학과 박사과정
†† 정 회 원 : 선문대학교 IT교육학부 전임강사
††† 종신회원 : 인천대학교 컴퓨터공학과 교수
논문접수 : 2013년 6월 12일
심사완료 : 2013년 7월 23일
* Corresponding Author : Min Byoung Joon(bjmin@incheon.ac.kr)

논문의 구성은 다음과 같다. 2장에서 악성페이지 탐지와 웹 수집에 대한 관련연구를 서술하고, 3장에서 웹 환경 변화에 따른 보안 상 문제를 제기한다. 4장에서 문제 해결을 위한 제안모델을 제시하고 5장에서 구체적인 스케줄링 방안을 기술한다. 6장에서 스케줄링에 대한 시뮬레이션 실험 결과에 대해 논하고 7장에서 결론을 맺는다.

2. 관련 연구

웹에서 악성사이트를 탐지하고 차단하기 위한 많은 연구가 진행되고 있다. 클라이언트와 서버 사이에서 트래픽이 발생했을 때 이를 가로채서 진단하는 수동적인 방법[1,3,4,5]과 능동적으로 악성사이트를 찾아 다니는 방법[2]이 있다.

트래픽이 발생되었을 때 수동적으로 진단하는 경우는 트래픽 수집 위치에 따라 서버측 웹방화벽 형태[3], 클라이언트 측 프록시[1,4], 브라우저 확장 형태[5]로 구분할 수 있다. 서버 측에서 트래픽을 수집하고 분석하는 경우[3]는 웹서버의 정상적인 동작 정보를 활용할 수 있다. 그러나 매쉬업의 경우 클라이언트가 여러 서버와 통신을 하기 때문에, 수집하는 서버 측에서 모든 트래픽을 수집할 수 없다. 클라이언트 측에서 수집하는 경우[1,4]는 다수의 클라이언트가 이용하는 불특정 사이트를 진단하여 클라이언트의 악성사이트로의 접속을 차단한다. 이때 별도의 장비 설치가 필요하고, 모든 클라이언트가 혜택을 볼 수 없으며, 차단 이외의 사후 조치를 취할 수 없다. 브라우저 확장[5]은 소프트웨어 컴포넌트로 별도의 장비 구성이 필요 없이 적용할 수 있으나 사용자 개인이 설치해야 한다.

능동적으로 악성사이트를 찾는 경우[2]는 탐지 대상을 수집하여 악성코드 삽입 여부를 진단한다. HoneyMonkey[2]는 악성사이트로의 트래픽 유입을 분석하여 사이트간 연결관계를 탐지하는 방법을 설계, 구현하였다. 탐지 대상을 사전에 선별하면 탐지 효율을 높이고 탐지 비용을 절약할 수 있다. [6]은 피싱사이트의 URL 특징을 회귀분석으로 모델링하여 탐지 대상의 사전 정보 없이 필터링하는 방법을 제시하였다. [7]은 악성 웹사이트의 특징을 학습하고 통계적 방법으로 URL을 분류하고 예측하여 악성사이트를 효율적으로 찾는 방법을 연구하였다. 구글은 정보검색을 위해 수집한 웹페이지를 시그니처 기반의 정적 분석, 브라우저 에뮬레이터를 통한 동적 분석, 가상화를 통한 시스템 영향 등을 진단한 결과로 안전브라우징 서비스 인프라를 구축하고 이를 오픈 API로 제공하고 있다[8].

웹 검색 분야는 웹을 지속적으로 수집하고 색인하는 문제를 다루어 왔다. 웹의 링크구조를 파악하고 중요한 페이지를 중복없이 빨리 수집하는 방법[9], 일시적인 페이지와 영구적인 페이지를 구분하여 적은 비용으로 페이지 변경을 예측하고 수집하여 최신 페이지로 유지하는 방법[10] 등이 연구되었다.

본 논문에서는 악성페이지 탐지와 웹 수집 기술을 이용하여 웹페이지의 특성에 따라 악성페이지를 능동적으로 탐지하고 지속적으로 감시하기 위해 진단 페이지를 선별하고 재수집하는 스케줄링 방안을 제안한다.

3. 문제제기

3.1 웹의 확장성

수많은 링크, 사용자 참여, 서비스의 연계로 웹은 확장되고 있다. 웹사이트의 안전성과 서비스 유지를 위해 웹사이트 내부의 보안 강화가 중요하지만 웹이 확장되었듯이 보안을 위한 진단 대상도 확장되어야 한다.

3.2 진단 대상의 선별

외부 링크를 몇단계 따라가면 진단 대상이 방대해지고 진단에 많은 자원이 소요된다. 그렇다고 무턱대고 축소하여서도 안된다. 해커들은 탐지를 어렵게 하고 차단되더라도 지속적으로 공격을 수행하기 위해 중간에 여러 경유지를 두기도 한다[11]. 이런 경우를 감안하여 진단 범위를 선별할 필요가 있다.

3.3 지속적인 진단의 필요성

웹은 지속적으로 변화한다. 매주 8% 새로운 페이지가 생성되고 1년에 80%가 사라질 정도로 역동적으로 변한다[12]. 또한 웹자원을 가리키는 주소로 쓰이는 URL은 동일해도 그 대상은 변화한다. 이전의 진단 결과가 현재의 안전을 보장하지 않는다. 지속적으로 링크 주소의 웹페이지를 수집하여 안전성을 확인해야 한다.

3.4 기존 연구의 한계

관련 연구에서 악성사이트를 탐지하고 차단하기 위해 검사 대상을 수집하는 기준에 따라 수동적인 방법[1,3,4,5]과 능동적인 방법[2]으로 구분하였다. 수동적인 방법은 사용자의 웹사이트 이용시 발생하게 되는 트래픽을 가로채서 접속하는 사이트가 안전한지, 악성코드가 삽입되어 있는지를 진단한다. 진단 작업이 개별 트래픽이 발생했을 때 실행되어 진단 후 종료된다. 이는 발견된 악성사이트와 알려진 공격을 차단하는데 적절한 방식이다. 능동적인 방법은 진단 대상의 수집과 진단이 사용자와 무관하게 일정한 스케줄에 의해 수행된다. 진단의 시작이 되는 시드(seed) URL로부터 링크된 사이트를 크롤링하며 악성 여부를 판단한다. 이는 기존 공격 기술을 이용하는 새로운 악성사이트를 찾는데 적절하다. 진단 대상이 방대하여 시드 URL을 선별하여 진단함으로 진단의 효율을 높이는 방법이 제안되었다. 이는 진단되지 않은 신규 사이트의 발견에 중점을 두고 있어 기존 정상적인 사이트와 진단을 마친 후 해킹으로 변조된 사이트는 진단 대상에서 제외된다. 새로운 악성사이트를 발견하는 것도 중요하지만 결국 정상사이트와 악성사이트 간에 연결이 발생할 때 실질적인 공격이 발생하게 된다. 따라서 본 논문은 정상적인 사이트를 지속적으로 감시하여 악성코드의 삽입이나 악성사이트로의 링크를 탐지하기 위한 스케줄링 방안을 제시한다.

4. 제안모델

4.1 진단 대상 분류

앞에서 제기한 문제를 해결하기 위해 확장된 진단 대상을 자체적으로 서비스하는 영역과 외부와 긴밀하게 연계되어 서비스되는 영역, 링크에 의해 연결 관계를 갖는 영역으로 나눈다. 각각을 내부사이트, 연계사이트, 외부사이트라고 명명한다.

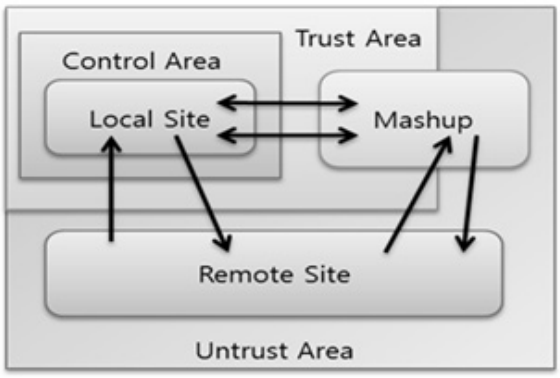


Fig. 1. Classification of Web Sites for Security Check

내부사이트는 보호하고자 하는 궁극적인 대상으로 서비스의 주체적인 역할을 수행하며 서비스 제공자 입장에서 로컬에 위치하여 관리와 통제를 받는 웹사이트를 의미한다. 내부사이트는 제어가 가능한 영역으로 직접 서비스를 제공하고 변경을 관리하기 때문에 보안에 대한 책임을 갖고 악성코드 발견 시 즉각적인 대응과 수정이 가능하다.

연계사이트는 매쉬업, 위젯, 광고 등으로 외부에 웹페이지, 이미지, 스크립트 코드 등이 존재하지만 단독으로 사용되기 보다 내부 사이트와 상호작용하며 시각적으로 내부사이트의 페이지와 같은 화면에 표시되는 웹사이트이다. 클라이언트 브라우저가 내부 웹사이트에 접속했을 때 자동으로 접속이 되며 출처에 대한 구분이 어렵고, 내부 사이트와 연계 사이트가 직접 통신하거나 클라이언트에 다운로드된 양쪽 자원이 상호 동작하며 통신한다. 연계사이트는 내부사이트와의 사전 협의에 의해 상호 참조하여 자유롭게 서비스가 제공되며 통제권한은 없다. 연계사이트에 보안사고가 발생할 경우 전체 서비스의 일부 기능이 오동작하거나 정지되는 등 영향을 받는다.

외부사이트는 넓은 의미로는 내부와 연계 사이트를 제외한 모든 원격지 사이트이며 좁은 의미로는 내부사이트와 링크로 연결된 사이트이다. 잠재적으로 모든 사이트가 링크로 연결될 수 있으며, 시간의 경과에 따라 링크가 삽입되고 유지되다가 삭제된다. 외부사이트는 내부사이트에서 간단한 클릭으로 브라우저의 페이지가 이동되거나 새창으로 접속된다. 외부사이트는 신뢰할 수 없는 영역으로 서비스 구현 시 의도하지 않은 단순한 링크가 대부분이다. 외부사이트로의 접속 시 민감한 데이터의 접근, 외부로의 정보 전달, 취약점을 노린 악성코드 설치 등을 파악하고 경고하거나 차단한다. 연결이 차단되어도 서비스 자체에 미치는 영향은 미비하다.

4.2 진단 대상 수집

진단 대상은 관련연구에서 다루었듯이 수동적인 방법을 이용한 수집의 한계로, 능동적인 방법으로 대상을 수집한다. 내부사이트는 시스템에 대한 접근권한이 있기 때문에 소스코드 스캐닝과 웹서버의 액세스 로그 분석을 통해 진단 대상 URL과 구현코드를 수집한다. 구조를 알 수 없는 외부사이트와는 달리 내부사이트는 복잡한 구조 파악과 연결이 끊어진 페이지까지 발견이 가능하다. 연계사이트는 상호 연계를 위해 협의된 규칙의 범위 내에서 예측과 수집이 가능하지만 연계 기능 이외의 페이지는 외부사이트와 동일하게 취급된다. 외부사이트 대상은 내부사이트에서 발견된 링크에서 추출된다. 외부사이트 링크 정보는 추가, 수정, 삭제되며 목록으로 관리된다. 대상 URL은 주기적으로 크롤러(crawler)가 능동적으로 접속하여 페이지를 다운로드함으로 수집된다.

4.3 진단 순서 결정 요소

웹사이트의 악성페이지 탐지를 위한 수집은 일회성으로 끝나지 않는다. 진단 대상 페이지는 유동적으로 변하기 때문에 지속적으로 수집하여 진단해야 한다. 진단 대상을 내부에서 연계 사이트와 외부 링크 사이트까지 확장하였기 때문에 전체 대상을 진단하는데 상당한 시간이 소요된다. 진단 대상의 규모를 유지하면서 진단 주기를 줄이려면 많은 컴퓨팅 자원을 투입해야 하고 네트워크 대역폭의 과다 사용으로 사용자의 웹사이트 이용을 저해한다. 무엇보다 투입되는 비용이 진단 효과를 초과하게 된다. 그러므로 한정된 자원을 이용하여 진단 자체가 서비스에 영향을 미치지 않으면서 악성페이지를 적시에 탐지하기 위해 진단 순서를 조절한다. 페이지 특성에 따라 중요도가 높은 페이지는 자주 진단하고 중요도가 낮은 페이지는 가끔 진단한다. 이러한 진단 순서를 결정하기 위한 중요도 특성을 도출하여 안전성 확인, 위험성 의심, 진단 노후성으로 분류하였다.

1) 안전성 확인 요소

안전성 확인 요소는 사용자가 많이 접속하고 중요도가 높은 페이지를 우선적으로 빈번하게 진단하기 위해 도출하는 특성들이다. 사용자가 거의 이용하지 않아 접속 가능성이 희박한 페이지를 진단하는 것보다 가장 많이 이용하는 페이지를 진단하는 것이 안전성 확인 효과가 크다. 자주 변경되거나 새로 생성된 페이지, 연계사이트와 상호작용이 활발한 페이지, 접속량이 많은 페이지에 추가된 링크 등을 먼저 진단한다. 안전성 확인 요소로 사용할 수 있는 특성으로 인기도, 접근성, 최신성, 변경도, 결합도 등이 있다.

- 인기도: 사용자가 많이 요청하거나 접속할 가능성
- 접근성: 초기 접속 페이지로부터의 링크 거리

- 최신성: 아직 진단하지 않은 새로 발견된 페이지
- 변경도: 페이지의 변경빈도에 따른 변경가능성
- 결합도: 연계사이트, 외부사이트와의 상호작용 정도

2) 위험성 의심 요소

위험성 의심 요소는 악성 웹사이트 특성과의 유사도, 탐지 이력 정보 등을 이용하여 악성페이지 의심 정도를 산정하기 위해 이용되는 특성들이다. 도메인명이 피싱사이트와 유사한 패턴을 보이는 경우, 악성코드 유포지나 경유지로 이용된 도메인이나 서버 IP인 경우, 인접한 위치에 있는 경우, 수집 시 서비스가 불안정하여 다운로드 실패율이 높거나 일부 페이지로만 구성된 웹사이트의 경우도 악성코드를 내포할 위험이 높다. 위험성 의심 요소로는 의심도, 유사도, 인접성, 빈발성 등이 있다.

- 의심도: 악성사이트의 패턴 내포 정도
- 유사도: 유명 도메인명, 블랙리스트 도메인명과 유사
- 인접성: 악성 사이트와의 링크 거리
- 빈발성: 악성코드가 자주, 최근 발견된 정도

3) 진단 노후성 요소

진단 노후성 요소는 안전성 확인 요소나 위험성 의심 요소의 수치가 낮아 우선순위가 낮은 페이지가 일정한 시간 이내에 진단될 수 있도록 보장해 준다. 진단 이력 정보를 이용하여 마지막 진단 이후 경과 시간에 따라 우선 순위를 조절하여 일정 시간 범위 내에서 진단되도록 한다.

- 노후도: 마지막 진단시간으로부터 경과 정도

4.4 진단 방법

악성코드 진단은 지금까지 축적된 기술과 정보를 활용한다. 구글은 안전 브라우징 서비스를 제공하기 위해 시그니처를 통한 패턴 일치, 코드 정적 분석, 스크립트 엔진을 통한 실행, 가상화 환경에서의 동작 감시를 통하여 악성코드를 진단한다[8].

본 연구에서는 알려진 공격의 발견과 차단에 초점을 둔다. 클라이언트의 보안 수준을 강제하거나 보장할 수 없기 때문에, 클라이언트가 취약성을 내포하고 있다는 전제하에 클라이언트가 웹사이트 이용 시 접속 가능한 페이지의 안전성을 서비스 제공자 관점에서 점검하고 악성페이지를 발견하고 차단하는 것을 목표로 한다.

5. 제안 스케줄링

5.1 스케줄링 요소

클라이언트가 웹사이트 이용 시 접속하게 되는 웹페이지의 안전을 확인하고 악성페이지를 효율적으로 탐지하기 위해 진단 순서를 부여한다. 진단 순서를 결정하는 요소는 앞에서 서술했듯이 인기도, 접근성, 최신성, 변경도, 의심도, 유사도, 빈발성 등의 요소를 조합하여 도출할 수 있다. 본 논문에서는 사용자가 많이 이용하거나 악성코드가 내포된 것으로 의심되는 웹페이지를 자주 진단하는 것을 목표로 하여 인기도, 유사도에 초점을 맞추어 스케줄링 방법을 제안하고 실험하였다. 이번 장에서 진단 순서를 부여하기 위한 스케줄링 요소의 도출 방법을 기술한다.

1) 인기도

인기도는 사용자가 많이 접속하는 페이지에 높게 부여되며 인기도가 높은 페이지 내에 링크로 포함된 페이지도 사용자가 접속할 가능성이 높다. 내부페이지의 액세스 로그 분석을 통해 과거로부터 현재까지의 접속량을 도출한다. 지금까지 접속이 많은 페이지가 인기도가 높다. 인기도 계산 방법은 수식(1)로 표현된다. 로컬페이지 집합 L에 속하는 페이지의 인기도는 각 페이지의 접속수 AC를 최대 AC값으로 나누어 정규화한다. 로컬페이지가 아닌 경우는 PageRank[13] 알고리즘을 활용하여 로컬페이지에서의 링크 구조로 인기도를 도출한다. $\Gamma^-(p)$는 페이지 p를 가리키는, p로의 링크를 갖고 있는 페이지 집합이다. $\Gamma^+(p)$는 페이지 p가 가리키는, p내에 포함된 링크 대상 페이지 집합이다.

$$Rank(p_i) = \begin{cases} \dfrac{AC(p_i)}{\max\limits_{j \in L} AC(p_j)}, & \text{if } p_i \text{ is local page} \\ \sum\limits_{p_k \in \Gamma^-(p_i)} \dfrac{Rank(p_k)}{|\Gamma^+(p_k)|}, & \text{otherwise.} \end{cases} \quad (1)$$

2) 유사도

도메인 유사도는 위험성 의심 요소인 의심도, 인접성, 빈발성에 비해 상대적으로 쉽게 구축하여 계산이 가능하다. 정상사이트와 악성사이트 도메인 목록만으로 단순한 문자 비교 연산으로 유사도를 구한다. 해커는 정상사이트와 유사한 도메인으로 사용자를 속인다. 또한 악성사이트로 발견되어 차단될 것을 고려하여 도메인을 생성할 때 자동화하여 비슷한 도메인 명을 이용한다. 그러므로 유사도는 정상사이트나 악성사이트와 유사도가 높은 경우를 모두 위험성이 높다고 본다. 다만 정상사이트와 일치하는 경우는 유사도 계산 값이 1로 나오지만 위험성이 낮으므로 유사도 0으로 처리한다.

유사도 계산 방식은 수식(2)와 수식(3)으로 표현된다. 유사도 비교를 위해 도메인 문자집합 $S(d)$를 n-$gram$으로 생성한다. $Sim(d_i, B)$는 악성사이트 블랙리스트와의 유사도 값이며 블랙리스트 집합 B와 일치하면 유사도 1이 된다. 식(3)의 $Sim(d_i, W)$는 화이트리스트와의 유사도로 화이트리스트 집합 W와 일치하면 안전한 사이트의 도메인명이므로 유사도 0을 갖는다. 두 수식으로 도출한 값 중 큰 값을 최종 유사도로 부여한다.

$$Sim(d_i, B) = \begin{cases} 1, & \text{if } d_i \in B \\ \max\limits_{j \in B} \dfrac{|S(d_i) \cap S(b_j)|}{|S(di) \cup S(b_j)|}, & otherwise \end{cases} \quad (2)$$

$$Sim(d_i,W) = \begin{cases} 0, & \text{if } d_i \in W \\ \max\limits_{k \in W} \dfrac{|S(d_i) \cap S(w_k)|}{|S(di) \cup S(w_k)|}, & otherwise \end{cases} \quad (3)$$

3) 노후도

노후도는 최근 진단시간으로부터의 경과 시간을 의미한다. 노후도를 구하기 위해서 전체 페이지의 진단에 소요되는 예상 시간과 각 페이지별 최근 진단 시간이 필요하다. 이를 도출하는 방식을 수식 (4), (5), (6)으로 표현하였다.

대상 페이지 $P = \{p_1, p_2, p_3, \cdots, p_n\}$이고 페이지 p_i의 진단이 완료된 시간을 t_i라고 할때, 각 페이지의 진단 수행완료 시간은 $T = \{t_1, t_2, t_3, \cdots, t_n\}$이다. 이때 전체 페이지 진단에 소요된 시간 *Freq*는 수식(4)와 같이 계산되며, 페이지의 평균 진단 시간 *Tavg*은 수식(5)와 같다.

$$Freq = \sum_{t_i \in T} (t_i - t_{i-1}) = t_n - t_0, \ t_0 \text{ is start time} \quad (4)$$

$$Tavg = \frac{Freq}{|P|} \quad (5)$$

*Freq*는 모든 페이지를 1회 진단하는데 소요되는 시간이며 집합 *P*와 *T*가 유동적이기 때문에 *Freq*는 페이지의 변화에 따라 달라진다. 순차 진단을 할 경우 모든 페이지가 *Freq*의 시간동안 1회 진단되며 진단대상의 추가, 삭제가 없는 경우 페이지의 진단 횟수는 동일하다. 페이지의 진단노후도는 수식(6)과 같이 현재시간에서 과거 진단 수행시간의 차를 진단주기 *Freq*로 나눈 값으로 한다.

$$Age(p_i) = \frac{t_{cur} - t_i}{Freq} \quad (6)$$

5.2 스케줄링 요소의 조합

일반적으로 스케줄링은 효율성과 공평성을 만족해야 한다. 효율성은 목표한 효과를 가장 잘 달성하기 위한 방법을 제시한다. 그러나 목표한 효과에 반하는 대상은 무한정 처리를 기다려야 하기 때문에 불공평하게 된다. 효율성을 만족하며 공평성에 위배되지 않도록 스케줄링을 해야 한다.

공평성 측면에서 스케줄링 도출요소의 가중치를 부여하지 않고 노후도 *Age*만을 고려하면 모든 페이지를 순차적으로 진단하게 된다. 사용자에게 인기있는 페이지에서 악성코드가 발견되면 짧은 시간에 많은 사용자가 접속하게 된다. 효율성 측면에서 빨리 발견하여 차단하려면 인기도 가중치를 높여 스케줄링하면 된다. 수식(7)에서 인기도와 노후도의 가중치 합이 가장 큰 페이지를 선출하여 순서대로 진단한다.

$$\arg\max_i [\alpha \times Rank(p_i) + \beta \times Age(p_i)] \quad (7)$$

수식(7)에서 α는 인기도 가중치, β는 노후도 가중치다. α=0, β=1일 때 모든 페이지가 노후도로만 계산되어 동일한 횟수로 진단된다. 0<*Rank*()<1이고 α=β=0.5이면, 인기도가 최하값 0일 때 2번의 진단 주기 동안 최소 1회, 인기도가 1이면 1주기 동안 최대 2회 진단하게 된다. 인기도 가중치가 커질수록 페이지간 진단 주기의 편차는 증가하게 된다.

Fig. 2는 수식 (7)을 알고리즘 의사코드로 표현한 것이다. 전체 진단 페이지에 대한 인기도와 노후도의 가중치 합을 서로 비교하여 최대값에 해당하는 페이지를 선정하여 진단하고 진단 후 노후도는 0으로 초기화시킨다.

```
procedure SelectCheckPage()
  for i=1 to N do              // Check total page
    curval = rank(i)*0.5 + age(i)*0.5 // Sum weighted value
    if maxval<curval then
      maxval = curval          // Update maximum value
      maxpage = i              // Update page id
    end if
    age_increase(i)
  end for
  age_init(maxpage)
  return maxpage               // Return selected page id
end procedure
```

Fig. 2. Pseudo code of decision algorithm

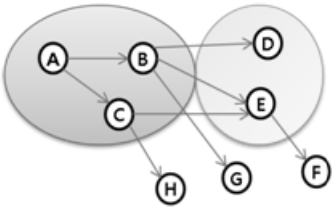
스케줄링 목적에 따라 인기도 이외의 스케줄링 요소를 선정하고 가중치를 부여하여 조합한다. 별도의 내부 보안이 갖추어진 사이트는 안전성 확보가 되었기 때문에 위험 요소에 해당하는 의심도, 유사도, 빈발성 등의 가중치를 높여 접근 가능한 외부의 악성페이지를 빨리, 많이 찾는 것을 목표로 스케줄링하는 것이 효과적이다. 또한 악성페이지 탐지 결과를 분석하여 요소들간 가중치를 조절하면 사이트 특성에 맞는 스케줄링의 도출이 가능하다.

5.3 스케줄링 효율

진단을 통해 악성코드 탐지 시 최선의 경우는 삽입 직전에 탐지하여 차단하는 것이고 최악의 경우는 진단 직후에 악성코드가 삽입되어 다음 진단 때까지 사용자가 접속하여 피해를 입는 것이다. 스케줄링하지 않고 순차 진단을 하는 경우는 악성코드 발생확률이 같을 때 삽입으로부터 진단까지 평균적으로 진단 주기의 절반의 시간이 소요된다. 스케줄링의 경우는 우선 순위가 낮은 페이지가 기아상태에 빠지지 않고 무한히 반복한다면 순차진단과 같겠지만 현실에서는 경우에 따라 우위가 달라진다. 다만 페이지별 사용자의 접속량을 분석해 보면 극히 일부 페이지에 집중되기 때문에 탐지 소요 시간과 탐지까지의 사용자 접속량을 곱한다면 피해 규모를 산정할 수 있다. 페이지별 공격 가능성이 동일하다면 페이지별 접속량과 평균 탐지 시간의 곱을 합하여 최적의 해를 구하여 스케줄링함으로 악성코드에 감염된 페이지의 접속량을 최소화할 수 있다.

5.4 시뮬레이션 사례

지금까지 서술한 웹사이트 진단 스케줄링의 동작 과정을 예를 들어 설명한다.



Page	A	B	C	D	E	F	G	H
Popularity	0.7	0.3	0.4	0.1	0.3	0.3	0.1	0.2
Suspicion	0	0	0	0.05	0.07	0.2	0.1	0.3

Fig. 3. Web Page Link Structure and its parameter settings

Fig. 3에서 페이지 A, B, C는 로컬사이트, 페이지 D, E는 연계사이트, 페이지 F, G, H는 외부사이트이고 각 페이지의 인기도와 의심도 값을 부여하였다. 이해를 위해 값이 고정된 것으로 가정한다. 모든 페이지의 진단 시간이 동일하다고 가정하고 8개의 페이지를 인기도 0.3, 의심도 0.3, 노후도 0.4의 가중치를 부여하여 계산하면 Table 1의 결과가 나온다.

Table 1. Decision of the Security Check Order of Web Page

Step	A	B	C	D	E	F	G	H	Choice
1	**0.21**	0.09	0.12	0.045	0.111	0.15	0.06	0.15	A
2	**0.21**	0.14	0.17	0.095	0.161	0.2	0.11	0.2	A
3	0.21	0.19	0.22	0.145	0.211	**0.25**	0.16	0.25	F
4	0.26	0.24	0.27	0.195	0.261	0.15	0.21	**0.3**	H
5	0.31	0.29	**0.32**	0.245	0.311	0.2	0.26	0.15	C

단계 마다 (노후도 증가)/(진단주기)*(노후도 가중치)에 해당하는 1/8*0.4=0.05 만큼의 값이 증가한다. Table 2는 가중치에 따른 페이지 선정의 변화를 나타낸다.

Table 2. Security Check Order of Web Page by Weighted-value

Popularity	Suspicion	Aging	Check Order
0	0	1	A B C D E F G H A B C D E F G H A B C D E F G H
0.3	0.3	0.4	A A F H C E A B G D F H C A E B G F H D A C E B
0.1	0.1	0.8	A F H C E B G D A F H C E B G D A F H C E B G D
0.5	0.1	0.4	A A A C A F E B H A G D C F A E B H A C G F D A
0.1	0.5	0.4	H F H A E G C D B F H A G C D F H B A E G C F H

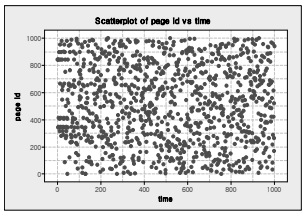
6. 실험 및 분석

6.1 실험 환경

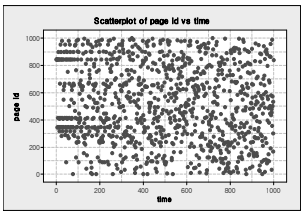
실험을 위해 운영중인 사이트의 페이지 정보와 로그를 수집하여 스케줄링 요소 중 일부 특성을 도출하여 모형화 하였다. 페이지 정보를 데이터베이스에 구축하고 진단 요소 가중치에 따른 연산을 수행하여 진단 대상을 선정하는 프로그램을 반복적으로 실행하여 시뮬레이션하였다. 악성사이트 의심도는 임의로 부여하고 의심도와 악성페이지의 일치 정도를 가정하고 추정치에 따라 악성코드를 삽입하여 시뮬레이션한 결과를 분석하였다.

6.2 실험 결과

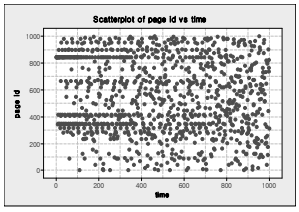
인기도와 노후도의 가중치 값을 변경하며 페이지를 진단하였다. 진단 결과 Fig. 4의 가로축은 시간 경과, 세로축은 페이지번호이다. 가중치 값 편차가 커질수록 인기도가 높은 특정 페이지에 진단이 집중되는 것을 확인할 수 있다. 인기도는 일부 페이지를 제외하고 대체로 낮기 때문에 인기도가 낮은 나머지 페이지는 노후도에 따라 일정 주기로 진단되었다.



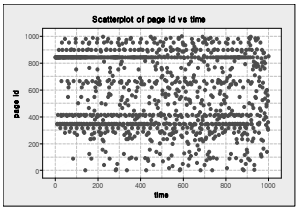
(a) Weight 0.5



(b) Weight 1



(c) Weight 1.5



(d) Weight 2

Fig. 4. Scatter plots of selected page by weighted popularity

Fig. 5은 악성사이트 의심도의 추정치를 부여하여 동일한 조건에서 악성코드를 삽입했을 때 삽입 후 탐지까지의 소요시간을 측정하였다. 가로축은 의심도 가중치, 세로축은 탐지 소요시간의 합이다. 의심도 추정치가 가장 높은 90%일 때 탐지 소요시간이 가장 낮게 나왔으나 가중치를 높여도 크게 향상되지 않았다. 반면 의심도 추정치가 낮은 경우, 즉 예측에서 벗어나는 악성페이지가 많은 경우는 가중치를 높였을 때 도리어 탐지 시간이 증가하였다. 시뮬레이션에서는 의심도 가중치 0.75, 노후도 1일 때 평균적으로 탐지 시간이 낮게 나왔으며, 악성페이지의 발견 패턴에 따라 달라질 수 있다.

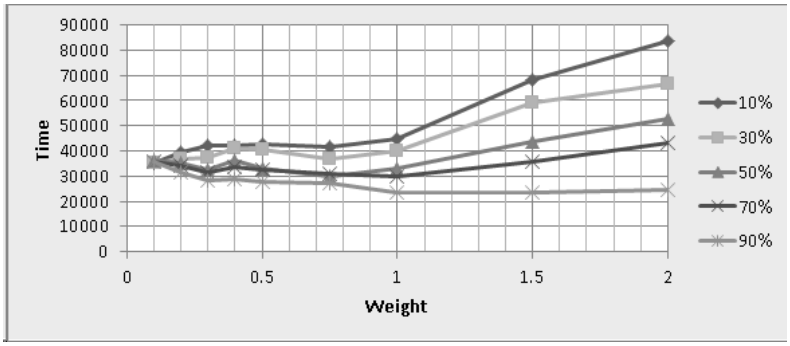


Fig. 5. Detection Delay Time based on Suspiciousness Estimation

7. 결 론

본 논문에서는 웹사이트의 안전적인 운영을 위해 안전성 진단을 사이트 내부뿐만 아니라 외부 링크까지 확장하여, 진단 대상을 선별하고 지속적으로 진단하여 악성페이지를 탐지하는 안전진단 스케줄링을 제안하였다. 진단 대상의 특징을 추출하고 수치화하여 진단 주기를 조절하는 스케줄링 방법을 제시하고 시뮬레이션을 통해 스케줄링 방법에 따른 진단 페이지의 분포 변화와 탐지 소요 시간의 변화를 확인하였다.

참 고 문 헌

[1] Moshchuk, Alexander, et al. “Spyproxy: Execution-based detection of malicious web content.” *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*. No.3. USENIX Association, 2007.

[2] Wang, Yi-Min, et al. “Automated web patrol with strider honeymonkeys.” *Proceedings of the 2006 Network and Distributed System Security Symposium*, 2006.

[3] Jovanovic, Nenad, Engin Kirda, and Christopher Kruegel. “Preventing cross site request forgery attacks.” *Securecomm and Workshops*, 2006. IEEE, 2006.

[4] Kirda, Engin, et al. “Noxes: a client-side solution for mitigating cross-site scripting attacks.” *Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 2006.

[5] Chou, Neil, et al. “Client-side defense against web-based identity theft.” *11th Annual Network and Distributed System Security Symposium (NDSS'04)*. 2004.

[6] Garera, Sujata, et al. “A framework for detection and measurement of phishing attacks.” *Proceedings of the 2007 ACM workshop on Recurring malcode*. ACM, 2007.

[7] Ma, Justin, et al. “Beyond blacklists: learning to detect malicious web sites from suspicious URLs.” *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.

[8] Rajab, M., et al. “Trends in circumventing web-malware detection.” Google, Google Technical Report, 2011.

[9] Cho, Junghoo, Hector Garcia-Molina, and Lawrence Page. “Efficient crawling through URL ordering.” *Computer Networks and ISDN Systems* 30.1, 1998, pp.161-172.

[10] Olston, Christopher, and Sandeep Pandey. “Recrawl scheduling based on information longevity.” *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.

[11] Stokes, Jay, et al. “Webcop: Locating neighborhoods of malware on the web.” *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2010.

[12] Ntoulas, Alexandros, Junghoo Cho, and Christopher Olston. “What's new on the web?: the evolution of the web from a search engine perspective.” *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004.

[13] Page, Lawrence, et al. “The PageRank citation ranking: bringing order to the web.” Technical report, Stanford Digital Library Technologies Project, 1999.

최 재 영

e-mail : jero@incheon.ac.kr
1999년 인천대학교 전자계산학과(학사)
2001년 인천대학교 컴퓨터공학과(석사)
2010년~현 재 인천대학교 컴퓨터공학과 박사과정
관심분야 : Web Security, Distributed System

김 성 기

e-mail : skkim@sunmun.ac.kr
1996년 인천대학교 전자계산학과(학사)
1998년 인천대학교 컴퓨터공학과(석사)
2006년 인천대학교 컴퓨터공학과(박사)
2006년~2009년 인천대학교 초빙교수
2009년~현 재 선문대학교 IT교육학부 전임강사
관심분야 : Computer & Network Security, Distributed System, Ubiquitous Computing

민 병 준

e-mail : bjmin@incheon.ac.kr

1983년 연세대학교 전자공학과(학사)

1985년 연세대학교 전자공학과(석사)

1991년 미국캘리포니아대학(UC Irvine) 전기및컴퓨터공학과(박사)

1995년~현 재 인천대학교 컴퓨터공학과 교수

관심분야 : Computer & Network Security, Distributed System