보안 이슈

AhnLab 보안 전문가의 심층분석! 보안 이슈 정보를 전해드립니다.

클라우드 침해 원인의 80%는 이 것, 해결책은?

AhnLab 2021-11-01

오늘날 클라우드는 과거 기대했던 것 이상으로 보편화되어 있다. 이제 사용자들이 직/간접적으로 생산해내는 방대한 데이터는 PC와 스마트폰을 넘어 클라우드에서 관리된다. 정보를 관리해야 하는 기업들은 기존보다 더 많은 데이터를 획득할 수 있는 대신 더 큰 관리 책임을 안게 됐다.

이처럼 새로운 기술적 & 문화적 배경을 조성한 클라우드를 위협하는 가장 큰 요소는 무엇일까? 이번 글에서는 클라우드의 성장에 따른 보안의 현주소를 조명하고, 클라우드를 안전하게 운영하기 위해 필수적인 사항들을 살펴본다.

기업들은 기존 데이터센터 혹은 온프레미스 환경에 있는 인프라를 클라우드로 이전하면서 크게 두 가지 방식을 활용해왔다. 많은 투자를 통해 '클라우드 네이티브(Cloud Native)' 방식으로 이전한 기업도 있고, 보다 빠른 전환을 위해 'Lift and Shift' 방식으로 이전한 경우도 있다.

새롭게 만들어지는 서비스의 경우는 논외이지만, 국내에서 대부분의 서버 애플리케이션은 'Lift and Shift' 형태로 이전되고 있다. Lift and Shift는 설치형 서버가 가상 서버로 바뀌는 부분에 대해서는 큰 어려움이 없지만, 기존과 완전히 다르게 바뀌는 네트워크 환경이나 클라우드 인프라 서비스의 경우 복잡한 설정이 다양한 형태로 존재한다.

새로운 환경으로의 기술적 & 문화적 이동

먼저 클라우드 환경에서의 네트워크는 부하를 다양한 방식으로 분산시키는 '논리적 네트워크'로 변경된다. 기존 방화벽은 클라우드 서비스에서 기본으로 제공하는 'Access-List' 방식의 방화벽을 사용할 수 있고, 좀 더 정밀한 방화벽이 필요한 경우 호스트(Host) 기반 방화벽을 활용할 수 있다.

침입방지시스템(IPS)의 경우, 인라인(In-line) 방식 호스트 기반 IPS를 사용해야 한다. 호스트 IPS는 'North-South' 방어 뿐만 아니라, 'East-West' 방어도 가능해, 기존 온프레미스에서 사용하던 IPS보다 더욱 정밀한 탐지와 차단이 가능하다. 또한, 클라우드 워크로드에 대해서도 기존과 동일 수준 혹은 그 이상의 보안을 확립할 수 있게 된다.

두번째로 클라우드 인프라 설정은 축적된 경험과 노하우를 요구한다. 단순하게 접근할 경우, 기본 설정을 사용하면서 시행착오를 겪을 수밖에 없다. 특히, 컨테이너(Container), 도커(Docker), 쿠버네티스(Kubernetes) 등은 클라우드의 핵심이자 클라우드 환경을 위해 존재하는 기술로, 데브옵스(DevOps)와 CI/CD(Continuous Integration/Continuous Deployment), 즉 지속적인 통합 및 배포 방식과 함께 공존해야 한다.

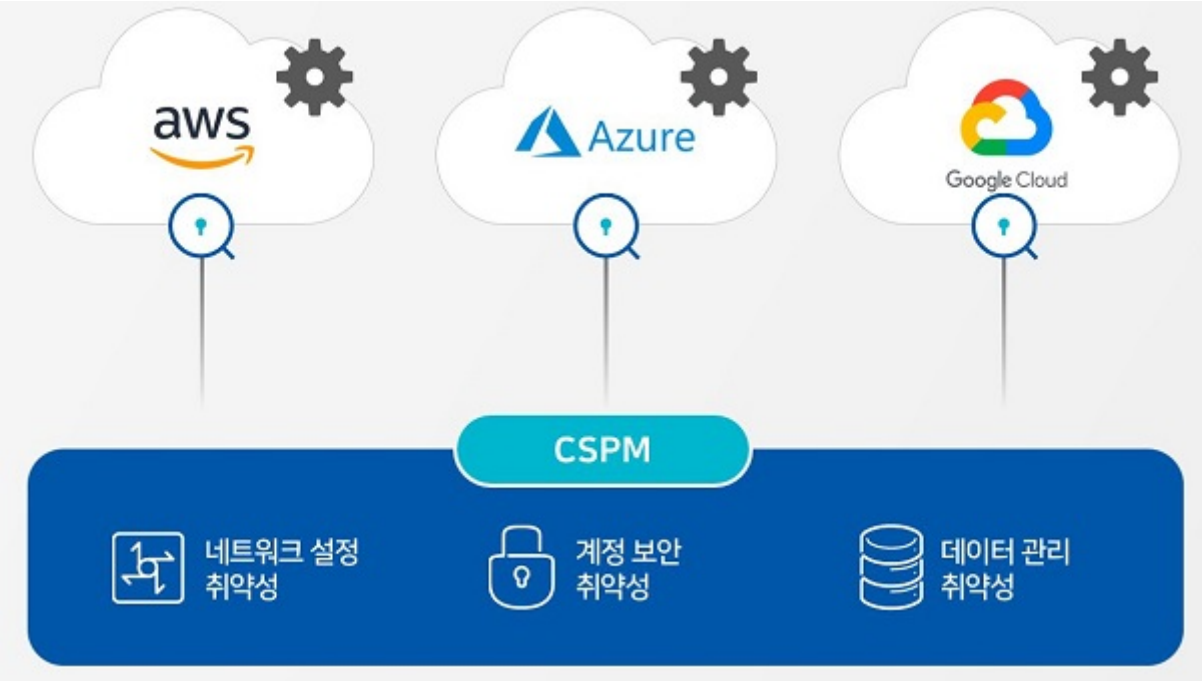
클라우드 위협의 본질과 CSPM의 역할

클라우드는 기술과 문화의 장벽이 높은 편으로, 초기 도입 시 여러 도전 과제를 마주하게 된다. 그리고, 해커들은 사용자가 미처 관리하지 못하는 취약한 부분을 노린다. 많은 자원을 투입해 방화벽이나 호스트 IPS 보안을 뚫기에 앞서, 일반적으로 실수가 많은 부분을 먼저 공략하는 것이다

특히, 클라우드로 처음 서비스를 이전하는 기업은 초기 구축 시, 클라우드의 여러 설정들을 어느 정도 열어놓고 이전을 진행하고, 실제 서비스를 시작할 때 설정을 그대로 방치한 채 운영하여 보안 사고로 이어지는 경우가 많다.

가트너의 보고서에 따르면 클라우드 보안 사고의 80% 이상이 설정 오류에 의한 것이며, 2025년까지 클라우드 보안 실패 사례 중 99%는 고객사의 잘못에 의해 발생할 것으로 예상된다. 특히 클라우드 서비스는 제공자와 사용자 간 '공동책임모델(Shared Responsibility Model)'을 근간으로 하기 때문에, 사용자가 클라우드 서비스 제공자(CSP)에 보안 사고의 책임을 묻더라도 대부분 경우 잘못을 입증하기 쉽지 않다. 이러한 보안 문제점들을 해결하기 위해 올바른 클라우드 설정 관리가 필수적으로 요구된다.

이 때문에 퍼블릭 클라우드의 설정 중 보안적 측면의 세부적인 관리를 수행하는 '클라우드 보안 형상 관리(Cloud Security Posture Management: CSPM)가 주목 받고 있다. CSPM은 잘못 설정된 네트워크 연결, 취약한 데이터 관리 설정 검토, 필요 이상으로 쉽게 접근 가능하도록 잘못 설정된 계정의 문제점을 탐색하고, 자동 관리가 가능한 모든 항목들에 대해 지속적인 탐지를 수행한다. 또, ISMS-P와 PCI-DSS 등의 컴플라이언스에 적합한 상태인지 여부도 파악할 수 있다.



[그림 1] CSPM의 역할

다음은 CSPM의 역할에 관한 예시이다.

AWS에서 루트(Root) 계정은 AWS의 모든 리소스를 제한 없이 접근할 수 있는 권한이 있으며, 일반적으로 사용하지 않도록 권장된다. 루트 계정을 최소한으로 사용하고 접근 관리를 위한 최소 권한을 부여하면, 자격 증명이 실수로 변경되거나 의도치 않게 공개될 위험을 줄일 수 있다. [그림 2]와 같이 AWS 관리 명령을 확인했을 때, JSON 포맷으로 출력된 내용 준 표시된 부분이 "0" 이 아니라면 루트로 접근 가능한 키가 존재한다는 뜻이다. 이 때는 AWS 관리 콘솔에서 키를 삭제해야 한다.

1
```
aws iam get-account-summary
```

2
```
{
  "SummaryMap": {
      "UsersQuota": 5000,
      "GroupsQuota": 100,
      "InstanceProfiles": 6,
      "SigningCertificatesPerUserQuota": 2,
      "AccountAccessKeysPresent": 1,
      "RolesQuota": 250,
      "RolePolicySizeQuota": 10240,
      "AccountSigningCertificatesPresent": 0,
      "Users": 27,
      "ServerCertificatesQuota": 20,
      "ServerCertificates": 0,
      "AssumeRolePolicySizeQuota": 2048,
  }
}
```

[그림 2] 루트 접근 키 현황

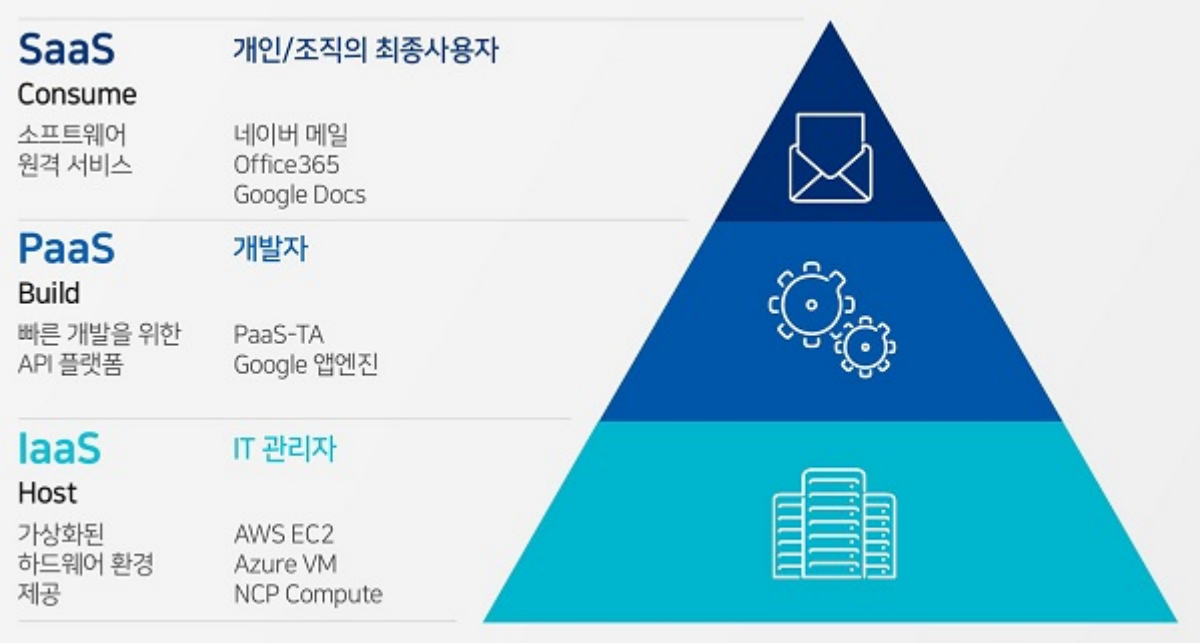
또 다른 예시로, 암호 정책은 보안 강화를 위해 패스워드 길이가 최소 14자 이상으로 구성되어야 한다. AWS 콘솔을 통해 확인해보면 [그림 3]과 같이 적절하지 못한 값으로 되어 있는 경우를 발견할 수 있다. 이 경우, 암호 정책을 변경하여 14자 이상의 패스워드 문자를 요구하도록 하는 것이 좋다.

[그림 3] 패스워드 길이 설정 현황

문제는 이처럼 기본적으로 검사해야 하는 항목들이 100가지 이상이며, 이를 항상 수동으로 진행하는데 한계가 있다는 것이다. CSPM은 이러한 검사와 문제점 해결을 체계적으로 그리고 주기적으로 수행해주는 유용한 보안 솔루션이다. 클라우드 설정은 아주 단순한 실수로도 큰 보안 사고로 연결되기 때문에 CSPM를 활용하여 보안 컴플라이언스를 확립하고, 취약한 설정을 상시 관리할 수 있는 체계를 구축해야 한다.

하드닝의 중요성: OS 설정 관리

클라우드로 제공되는 서비스는 대표적으로 서비스형 소프트웨어(Software-as-a-Service), 서비스형 플랫폼(Platform-as-a-Service: PaaS), 서비스형 인프라(Infrastructure-as-a-Service: IaaS)가 있다.



[그림 4] 클라우드 서비스 구조도

이 중 IaaS는 기존 데이터센터에서 사용하던 서버와 일대일로 대응되는 가상 서버이다. 국내 기업들은 클라우드의 서비스 중 IaaS를 많이 사용하는 편인데, 이는 기존 데이터센터에서 클라우드로 이전하기 가장 쉬운 방법이 IaaS의 가상 서버로 이전하는 것이기 때문이다. 앞서 언급한 공동책임모델을 기준으로, IaaS는 하드웨어와 네트워크를 제외한 나머지 모든 영역을 기업이 설정하고 운영해야 한다.

IaaS를 사용하는 기업은 운영체제(OS)를 선택하고 IaaS에 탑재해 운영하게 된다. 클라우드에서 사용되는 OS는 레드햇(Redhat), 우분투(Ubuntu), 아마존 리눅스(Amazon Linux), 오라클 리눅스(Oracle Linux) 등 여러가지가 있고, OS의 설정 관리 또한 기업의 몫이다. 이 부분에서도 잘못된 설정 관리로 인해 보안 사고가 발생하곤 한다.

다시 간단한 예시를 들어본다.

서버에 원격으로 접속하는 경우 대부분 SSH(Secure Shell)를 활용해 보안성을 높이려 한다. 하지만 SSH 서버의 설정이 [그림 5]와 같이 되어 있는 경우, 루트 관리자의 비밀번호가 노출되었을 때 시스템이 탈취당할 위험이 있다. 이처럼 취약한 보안 설정이 발견되면 관리자는 명령어를 통해 루트 계정의 SSH 서버 로그인을 차단해야 한다.

[그림 5] SSH 서버 보안 설정

ISMS-P 인증이나 CIS 벤치마크 등의 컴플라이언스 지침서에서도 잘못된 설정을 체계적으로 방지하기 위해 위와 같은 OS 설정을 금지하는 것으로 정의하고 있다.

하드닝의 중요성: 오픈소스 설정 관리

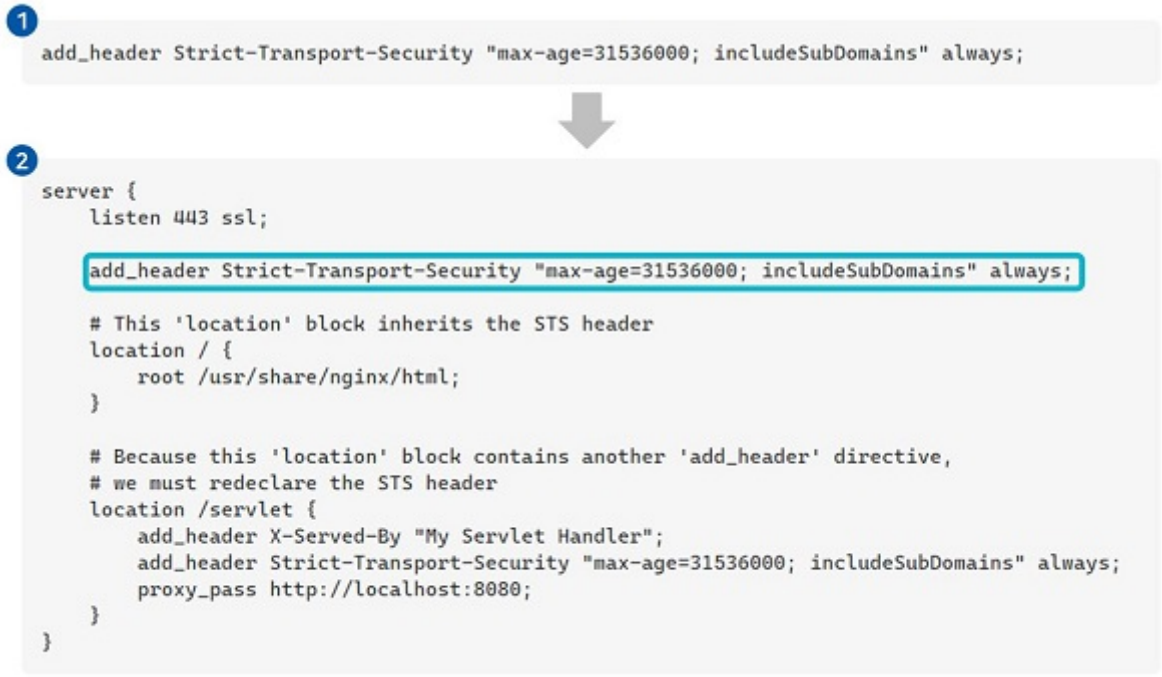
클라우드에서는 다양한 OS 뿐만 아니라 수많은 오픈소스가 사용된다. 보통 특정 소프트웨어가 필요하면 제일 먼저 오픈소스가 있는지를 찾아보게 되며, 이미 존재하는 오픈소스를 사용하지 않고 새로 개발하는 경우는 거의 없다고 봐도 무방하다. Nginx, MongoDB, 쿠버네티스 등은 범용적인 오픈소스로, 설정 또한 다양해 보안이 취약하지 않도록 주의를 기울여야 한다.

오픈소스의 취약한 설정에 대한 예시를 알아보자.

최근 대부분의 웹사이트들은 HTTPS 암호 통신을 사용한다. 웹서버는 공인된 인증서를 탑재하고 있어 브라우저가 HTTPS 접속 수행 시 인증서의 유효함을 확인한다. 하지만 해커는 사용자와 웹서버 중간에서 '프로토콜 다운그레이드(Protocol Downgrade)' 공격 등을 통해 정상 세션을 우회할 수 있고, 두 주체 간 통신을 가로채는 '중간자(Man in the Middle) 공격'을 수행해 사용자의 통신 내용을 모두 열어 볼 수 있다.

최근에는 이러한 상황을 방지하기 위해 HSTS(HTTP Strict Transport Security)라 일컬어지는 상위 레이어 보안 프로토콜이 대부분의 브라우저와 웹서버에 탑재되어 있다. 이 경우 웹서버는 브라우저의 HTTPS 접속을 강제하고, 프로토콜의 다운그레이드를 방지하며 해당 설정을 특정 기간 동안 유지하도록 하여 중간자 공격을 봉쇄한다.

HSTS 프로토콜은 웹서버의 설정에 의해 활성화된다. 반대로 말하면, Nginx 서버가 [그림 6]과 같이 설정되어 있지 않으면, HSTS 프로토콜이 활성화되지 않고 공격에 노출될 수 있기 때문에 설정을 변경해야 한다.



[그림 6] Nginx 서버 설정 관리

문제는 앞서 CSPM의 예시와 같이 오픈소스의 점검 항목이 Nginx만 해도 100개가 넘는다는 것이다. 마찬가지로 수동 관리에 한계가 있는데, OS와 오픈소스 등 시스템 설정을 효과적으로 관리할 수 있도록 하는 솔루션이 바로 '하드닝(Hardening)'이다.

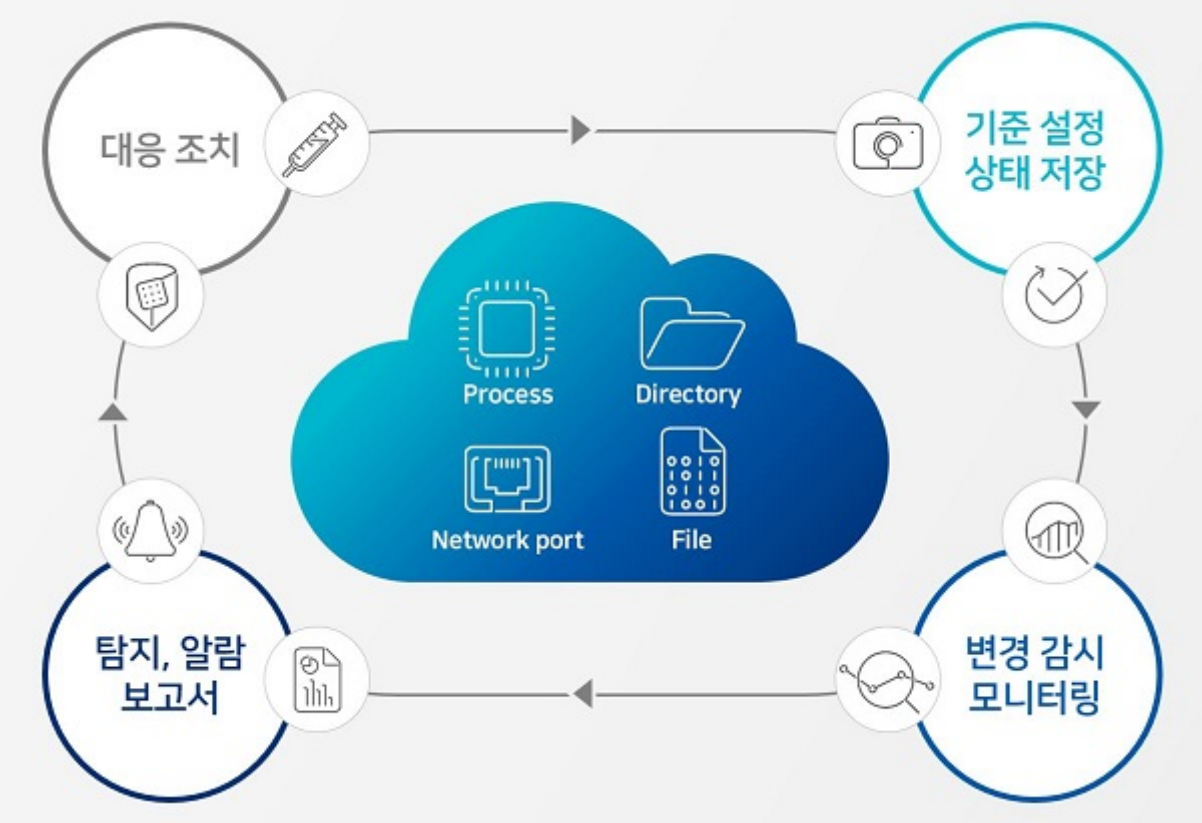
클라우드 워크로드 보안 플랫폼(Cloud Workload Protection Platform)에 탑재되는 하드닝은 OS에 최소한의 권한을 부여하고, DB 등 서비스 권한의 세밀한 조정, 역할 분리, 온디맨드(On-Demand) 허가 체계 등 다단계 방어 전략 수립을 지원한다. 또한, 사용하는 OS 및 모든 오픈소스의 설정 값이 보안 측면에서 적절한지 여부를 주기적으로 스캔한다.

클라우드와 오픈소스의 사용이 확대되고, 점검 항목이 많아지면서 위험 요소도 필연적으로 늘어날 수 밖에 없다. 클라우드를 향한 위협을 사전에 방지하기 위해서는 클라우드 인프라의 보안 위험 관리를 위해 CSPM을 적용하고, CWPP의 하드닝을 활용해 클라우드 워크로드의 서버 OS 및 다양한 오픈소스의 위험 요소를 지속적으로 관리해야 한다.

시스템 변화를 관리하는 무결성 모니터링

CSPM과 하드닝은 클라우드 보안의 핵심이지만, 보안 단계로 볼 때 사전 방어 체계이며 100퍼센트 완벽할 수는 없다. 추가적으로, 보안성 강화를 위해 시스템 변경이 언제, 어디에서 발생했는지 확인하는 사후탐지 체계 또한 굉장히 중요하다. 이 역할을 하는 것이 바로 '무결성 모니터링(Integrity Monitoring)'이다.

무결성 모니터링은 OS, 애플리케이션, 파일 뿐만 아니라 필수 프로세스와 네트워크 포트의 변경이 발생할 경우, 이를 탐지하고 사용자에게 알려준다. 무결성 모니터링의 원리는 정상 상태의 기준을 설정하고, 상태를 기억한 뒤 실시간으로 모니터링하면서 변경 상태를 감시하는 것으로 이해하면 된다.



[그림 7] 무결성 모니터링 개념도

구체적으로 설명하면, 시스템에서 서비스를 위한 네트워크 포트가 열려 있는 경우, 특정 상황에서 추가적으로 열리는 상황이 발생하는지 여부를 감시한다. 또, 정상적인 서비스를 위한 프로세스 이외에 알려지지 않은 프로세스가 있는지 모니터링하고, 특정 파일들이 예정되지 않은 시간에 불필요하게 변경되거나, 디렉토리 내의 파일들이 변경 또는 갱신되지 않는지 여부 등 시스템 변화를 점검한다.

모니터링이 필요한 네트워크 포트, 프로세스, 파일, 디렉토리는 OS 및 오픈소스마다 상이하고, 때로는 수백 개의 파일을 모니터링 해야하는 경우도 있다. 아울러, OS와 오픈소스는 계속 변화하기 때문에 무결성 모니터링의 자동화된 감시 체계를 통해 주기적/실시간 감시가 반드시 필요하다. 참고로 ISMS-P, PCI-DDS 등 주요 컴플라이언스 및 보안 프레임워크에서도 필수 보안 제어 요구사항을 해결하기 위한 효과적인 방법으로 무결성 모니터링을 강조하고 있다.

Agent Based vs Agentless

CSPM, 하드닝, 무결성 모니터링은 사용 방식에 따라 설치형인 에이전트(Agent) 기반과 비설치형인 에이전트리스(Agentless)로 나뉜다.

[그림 8] Agent Based vs Agentless 비교

에이전트 기반 보안 솔루션은 변경사항을 실시간으로 모니터링 할 수 있다는 것이 장점이다. 모니터링되는 호스트에 에이전트가 설치되기 때문에 OS와 오픈소스의 활동을 실시간으로 모니터링해 내용을 기록할 수 있다.

또, 다른 장점은 호스트를 개방할 필요가 없다는 것이다. 에이전트리스 방식으로 시스템 파일을 모니터링하기 위해서는 관리자 권한으로 개방되어야 하며, 이는 새로운 취약점이 발생할 수 있다는 뜻이기도 하다. 이 밖에 변경 사항에 대해서만 업데이트를 제공할 수 있다는 점에서 에이전트리스 방식에 비해 이점이 있다.

에이전트리스 솔루션은 변경 사항이 식별되었는지 여부를 판단하기 위해 전체 체크리스트를 실행해야 한다. 또한, 정교한 스크립트를 사용하더라도 결과를 추출할 때 호스트와 네트워크에서 상당한 자원을 소모할 수 밖에 없다.

반면, 호스트에 에이전트 소프트웨어를 배포할 필요가 없다는 것은 확실한 장점이다. 쉽게 말해, 언제든지 실행하고 결과를 얻을 수 있는 '신속한' 솔루션이라는 의미다. 검사를 수행할 대상 호스트가 다양하거나, 에이전트를 설치할 여유 공간이 부족한 경우 혹은 빠른 스캔과 결과를 필요로 할 때에는 에이전트리스 방식이 적합하다.

에이전트 기반과 에이전트리스 방식은 서로 장단점이 분명하기 때문에 사용 환경에 따라 적절한 솔루션을 선택하는 것이 중요하다. 일반적으로 CSPM은 에이전트리스 방식을 사용하는 것이 적합하고, 하드닝과 무결성 모니터링은은 에이전트 기반으로 설치해 운영하는 것이 안전한 보안체계 구축에 효과적이다.

결론: 클라우드 보안의 열쇠는 올바른 설정이 쥐고 있다

기업들은 보안 위협이 고도화되고 멀티 클라우드, 하이브리드 클라우드, 클라우드 네이티브 애플리케이션 등 새로운 비즈니스 서비스가 진화하는 환경을 마주하고 있다. 이 같은 상황에서 설정 오류와 사소한 실수로 인

한 대형 보안 사고를 방지하기 위해서는 이상 여부를 사전에 식별하고 시스템 변화를 실시간으로 감시하는 보안 체계 수립이 필수적이다.

CSPM, 하드닝, 무결성 모니터링을 적재적소에 활용하여 다양한 위협을 탐지하고 침해지표를 식별해 보안 체계를 확립하고 안전한 클라우드 운영 환경을 조성하길 바란다.

AhnLab 클라우드개발실 노영진 상무