저희 리눅스는 가장 많이 쓰이는 배포판 중 하나인 Ubuntu를 기반으로하여 제작하였습니다.

일반적으로 사용하는 Ubuntu server나 desktop이 아닌 Ubuntu Base라는 루트 파일시스템을 이용하여 커널과 여러 가지 패키지를 구성하고 저희가 원하는 대로 리눅스를 구축하였습니다.

리눅스 배포 파일은 ova 파일 형태로 제공될 것이며, 여기에 저희가 구현한 트레이닝 기능과 보안 점검 기능이 내장됩니다.

보통 iso로 제공되는 여러 배포판 이미지들은 다운로드 받은 이후 설치를 하는데 많은 시간이 소모됩니다.

또한 여러 가지 설정을 설치 도중에 하게 되고, 설치가 완료된 이후, 업데이트도 적지 않은 시간이 걸립니다.

하지만, 저희의 리눅스는 ova 파일로 제공되기 때문에 다운로드 받고 실사용까지 걸리는 시간이 1~2분 내외입니다.

보안 입문자들 중에는 리눅스를 처음 사용하는 유저도 있을 것이기 때문에, 설치 작업의 단축과 시간 절약이 장점이라고 할 수 있습니다.

또한, CUI 환경에서 진행되는 트레이닝의 특성상 실행하는 환경에 따라 텍스트 출력이 다를 수 있습니다.

저희 리눅스에서는 BoB-Linux의 일정한 해상도와 글씨 크기에 맞춰 텍스트를 구성했기 때문에 사용자가 최대한 보기에 좋게 가독성을 주어 트레이닝 기능을 구현하려고 하였습니다.

///////////////////////////////////////////////////////////////////

다음으로는 프로젝트를 진행하며 1차 발표 이후와 달라진 방향을 말씀드리겠습니다.

먼저 1차 발표 피드백 중 트레이닝 기능에서 너무 많은 방대한 내용을 다루는 것 같다는 말씀이 있었습니다.

저희도 그 점을 수용해서 다양한 분야들에 대한 학습을 제공하자는 의도였던 분야별 학습을 다른 방향으로 돌리기로 했습니다.

그래서 좀 더 보안 입문자에게 도움이 될 수 있는 내용을 담아 보안 관련 로드맵을 제공하고, BoB Linux 및 우분투 계열의 리눅스를 사용하는 보안 입문자에게 제공할 수 있는 보안 점검 프로그램을 만들기로 했습니다.

물론 시중에서도 많은 보안 점검 프로그램이 존재합니다.

대표적으로 nessus나 lynis와 같은 프로그램이 있지만, 저희는 입문자들이 보안 점검을 스스로 진행하면서 다양한 지식을 습득할 수 있는 방향으로 나아가기를 원했습니다.

이러한 목표에 맞추어 보안 점검 프로그램을 제작 중입니다.

마지막으로, GUI 환경에서 CUI 환경으로 변경하였습니다.

저희가 처음 생각하기로는, 초심자가 사용하기 좋은 환경은 아무래도 윈도우와 같이 익숙한 GUI 환경이라고 생각했습니다.

하지만, 터미널에서 사용자에게 입력을 받아 그에 대한 검사를 진행하고 결과를 출력하는 트레이닝 기능의 특성상 굳이 GUI에서 사용할 이유는 없다고 판단하였습니다.

또한 윈도우에 익숙해져 있는 사용자에게 GUI 환경의 리눅스를 사용하게 하는 것이 과연 입문자를 위한 일인가? 라고 생각해보았을 때, 꼭 그렇지 만은 않다는 결론을 내렸습니다.

오히려 리눅스라는 새로운 환경에 있어서 터미널이라는 중요한 요소에 빠르게 익숙해질 수 있도록 하고,

GUI 환경과는 다른 터미널만의 매력을 느끼게 해주는 것이 더 좋다는 판단이었습니다.

다음 장부터는 BoB Linux에 들어갈 기능들을 하나씩 말씀드리곘습니다.

///////////////////////////////////////////////////////////////////////////

우선 BoB Linux와 트레이닝 기능입니다.

///////////////////////////////////////////////////////////////////////////

처음에는 BoB Linux의 목적 중 하나를 일상적으로 실사용이 가능한 리눅스로 잡았습니다.

하지만, 그러한 목적을 가진 리눅스는 이미 너무 많이 존재하고, 그것들과 별다른 점이 없다고 보았습니다.

그래서 BoB Linux는 저희가 만든 트레이닝 프로그램 및 보안 점검 기능과 시너지를 낼 수 있는 하나의 환경으로 만드는 것으로 방향을 바꾸었습니다.

우선, BoB Linux에서는 트레이닝 프로그램을 극대화 시키기 위한 노력을 하였습니다.

앞서 말씀드린 것처럼 해상도와 글자 크기에 맞추어 트레이닝 기능을 구성하였습니다.

가독성이 중요한 트레이닝 기능 특성상 저희가 의도한 대로 글씨가 출력되는 것을 원했기 때문입니다.

또한 뒤에서 설명드릴 보안 점검 프로그램과의 연계를 위해서 일부러 취약한 환경으로 구성하였습니다.

처음에는 다양한 정책을 통해 리눅스 자체의 보안을 강화하는 식으로 진행하려 했지만,

실사용이 아닌 보안입문자를 위한 교육용 리눅스기 때문에 사용자가 배우기에 용이한 환경으로 구축하려고 하였습니다.

보안 점검 체크리스트를 통해 보안을 강화하는 방향으로 조치를 할 때 와는 정 반대의 작업으로, BoB Linux에서 안전하다고 판단되는 점검 내용에 대해서 의도적으로 취약하게 구성하는 방법을 취했습니다.

이를 통해 사용자에게 최대한 다양한 항목의 보안 점검 실습을 가능하게 하였습니다.

또한, BoB Linux에서는 프레임 버퍼를 이용한 방법으로 Putty나 GUI 환경 없이 로컬에서 한글을 입력하고 출력할 수 있습니다.

PuTTy를 사용하면 한글 입출력을 할 수 있지만, 저희가 대상으로 삼는 것은 보안 입문자기 때문에 최대한 편의성을 제공하여 별다른 설정 없이 곧바로 리눅스 로컬에서 한글 사용을 가능하게 하고 싶었습니다.

이를 위해서 프레임버퍼 터미널을 이용하여 한글 출력을 가능하게 했고,

uim 입력기를 적용한 후 키를 매핑하여 입력에서 한/영 전환을 가능하게 했습니다.

또한 본래 fbterm은 root 권한을 가지고 있어야 사용이 가능하지만, 권한 관련 설정을 통해 일반 사용자가 쓸 수 있게 하였고, 터미널에서 스크롤이 먹히지 않는 Key map 문제도 해결했습니다.

그리고 초기에 사용자가 원하는 계정으로 사용할 수 있게끔, 쉘 스크립트를 구성하여

간단하게 원하는 계정 이름과 비밀번호로 사용할 수 있도록 하였습니다.

///////////////////////////////////////////////////////////////

다음으로는 트레이닝 기능 부분입니다.

리눅스는 Windows와 같은 환경에 익숙한 입문자들에게 새롭고 어려운 운영체제 일 수 있습니다.

그래서 보안입문자가 리눅스에 익숙해질 수 있도록 명령어들을 직접 입력하며 학습하게 해주는 방향으로 트레이닝 기능을 제공합니다.

또한 보안 입문자들을 위해 보안이라는 분야에 관련된 커리큘럼에는 어떤 것이 있는지, 어떤 직업군이 있는지, 어떻게 보안 분야가 세부적으로 분류되는지, 또 분류된 것에서 어떤 기법들이 있는지 등을 세세하게 알려주는 보안에 대한 로드맵을 구성할 예정입니다.

그리고 앞서 말씀드린 것처럼 트레이닝이 실행되는 리눅스와 연계하여 시너지를 내는 방향으로 진행하고 있습니다..

////////////////////////////////////////////////////////////////////////

다음 장에서는 보안점검 프로그램에 대해서 설명드리도록 하겠습니다.

////////////////////////////////////////////////////////////////////////

보안 점검 프로그램은 리눅스에 대한 보안 점검 기능을 제공합니다.

동작 방식은 리눅스 관리 및 설정에 관련된 취약점을 KISA에서 발간한 가이드라인을 기반으로 보안 상태를 점검하게 됩니다.

이 과정에서 안전 / 경고와 같이 점검 결과를 알려주고, 이에 대한 조치 사항을 알기 쉽게 한글로 제공합니다.

사용자는 조치 사항을 따라 스스로 리눅스의 설정을 건드려보고 이를 통해 보안에 대한 여러 가지 사항에 대해 배울 수 있게 됩니다.

////////////////////////////////////////////////////////////////////////

또한, KISA의 가이드 문서는 주로 서버에 사용되는 Redhat 계열의 CentOS에 맞추어 서술되어 있습니다.

BoB Linux는 Ubuntu 계열이므로, 둘 사이의 차이로 인해 호환되지 않는 점검 조치 방법이 존재합니다.

예를 들어 패스워드 복잡성 설정 항목에서 KISA의 매뉴얼은 system-auth 파일을 통해 정책을 설정한다고 제시되어 있으나 ubuntu은 common-auth 파일을 통해 정책을 설정합니다.

저희가 구현하는 보안 점검 프로그램은 이러한 점을 반영하여 데비안 및 우분투 환경에 알맞는 점검 조치를 제공합니다.

////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////

위의 캡쳐는 저희가 구현 중인 보안 점검 프로그램의 출력 화면입니다. U-14는 취약점 번호이며, 이에 대한 다양한 사항에 대해 검사합니다.

그리고 검사한 항목에 대한 점검 사항을 출력한 후에 이에 대한 설명을 덧붙입니다.

앞에서 말씀드렸듯이, 리눅스를 취약한 환경으로 구축하여 최대한 다양한 부분에 대해서 보안 점검 실습을 수행할 수 있도록 제공할 것입니다.

보안점검 프로그램의 사용 방법은 보안 점검을 계정, 파일, 서비스, 디렉토리 등으로 파트를 나누어 사용자가 점검할 파트를 선택하면 이에 대한 목록이 제공되고, 넘버링된 취약점 번호를 입력하여 해당 부분에 대한 보안 점검을 실시합니다.

그리고 해당 항목에 대한 결과와 조치 방법에 대해 설명하여 사용자가 스스로 조치할 수 있도록 합니다.

//////////////////////////////////////////////////////////////////

마지막으로 최종 발표까지 남은 일정에 대한 진행 방향을 설명드리고 마치겠습니다.

//////////////////////////////////////////////////////////////////

리눅스 배포판 ova 파일을 배포하기 위해서 담당 멘토님께서 제공해주신 서버를 사용할 것입니다.

지금 생각하고 있는 것은 gitlab으로 배포를 하거나, 간단하게 Web 서버에 링크를 첨부하여 배포할 예정입니다.

이렇게 해서 트레이닝 기능과 보안 점검 프로그램을 내장한 리눅스 배포판이 저희의 최종 산출물이 됩니다.

//////////////////////////////////////////////////////////////////

이것으로 Black Moon 팀 발표를 마치겠습니다. 감사합니다.