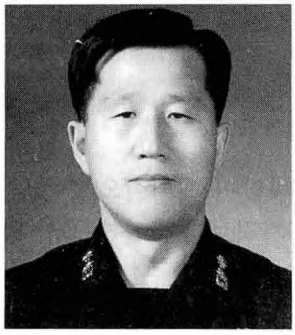
컴퓨터 바이러스 방해책 (2)

- 새로운 형태의 전자전 -



權 寧 根
공군전투발전단, 이학박사

오늘날의 새로운 군사 전자 시스템 기술은 여러 각도에서 유익한 점이 많이 있지만, 불행스럽게도 컴퓨터 바이러스에 의한 시스템의 침투 가능성을 높여 놓았다. 그러므로 향후 상업용 통신과 컴퓨터 시스템 기술을 설계 및 운영할 때는 이러한 바이러스들의 치명적인 능력들을 고려해야 한다

• Front-Door Coupling

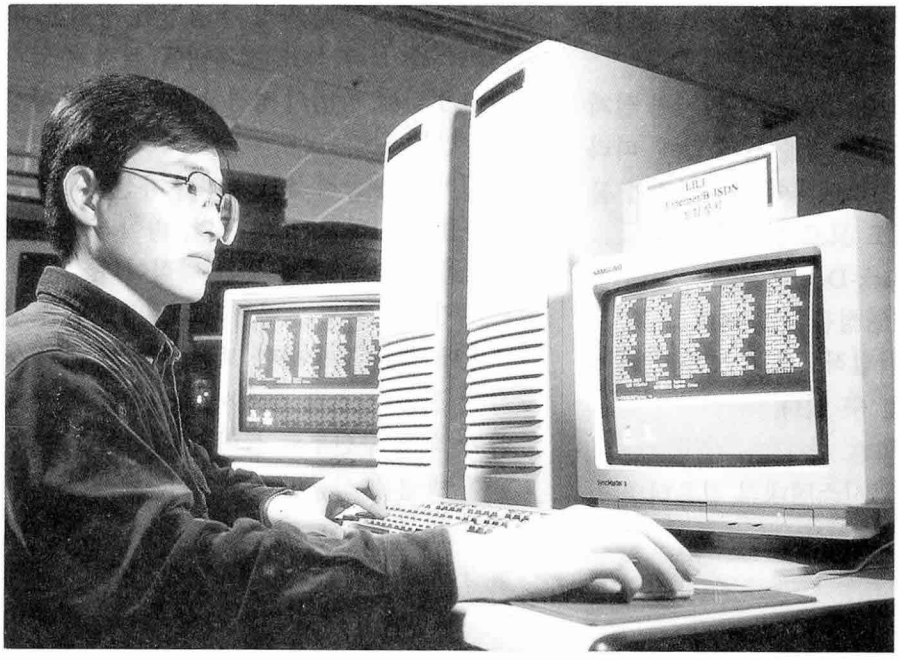
목표 시스템의 전파 매체(Media)를 이용하여 목표물에 접근하는 방식이다. 예를들면 전술(Tactical)작전에서 일반적으로 사용되는 라디오에의 Front-Door Coupling은 라디오 안테나와 수신 전자부분으로 날라오는 전자파를 사용해서 이루어진다.

C^3I 체계로의 Front-Door Coupling은 바이러스를 C^3I 체계의 자료 및 통제를 위한 통신 Link를 통해 목표물안에 주입시키며, C^3I의 수신부는 이러한 바이러스를 일상의 자료로 간주하여 처리하게 되고, 따라서 바이러스가 C^3I 체계에 이식 되는 것이다.

여기서 바이러스는 바이러스에 감염될 수 있는 모든 체계로 전파된다. 비록 대부분의 C^3I 체계들에 있어 이들의 방어 기제를 침투하기가 쉽지는 않지만, CVCM체계는 이러한 C^3I 체계의 방어가 가장 허술한 부분에 접목될 수만 있으면 된다.

대부분의 ECCM은 Jamming과 같은 방해전파에 대해 반응을 하도록 설계되어 있다. Jamming이 감지된 순간 수신부는 Coding을 증가시키거나, 방해 신호가 수신부와 연계되는 것을 막기위한 기술들을 수행하게 된다.

전자 방해책(Electronic Counter Measure)



◀기본적으로 CVCM은 목표 방어부의 가장 취약한 통신 Link 부분을 공격하면 되는 반면, ECM은 수신 방어부의 가장 강력한 통신 Link 부분을 무력화 시킬 수 있어야 한다

은 상대방 수신부의 ECCM을 무력화 시킬 수 있을 때에만 효과가 있다.

CVCM 체계는 CVCCM(Computer Virus Counter-Counter Measure)을 오직 한번만 극복하면 된다.

일단 바이러스가 이식되면, 수신부의 Mode가 바뀐다해도 바이러스의 효과는 지장을 받지 않는다.

기본적으로 CVCM은 목표 방어부의 가장 취약한 통신 Link 부분을 공격하면 되는 반면, ECM은 수신 방어부(Receiver's Defences)의 가장 강력한 통신 Link 부분을 무력화 시킬 수 있어야 한다.

• Back-Door Coupling

Back-Door Coupling은 목표 시스템이 아닌 다른 시스템의 통신 매체(Media)를 통해서 목표 시스템에 접근해 가는 기술로 정의될 수 있다.

CVCM 체계가 목표 시스템에 접근하면서 경유할 수 있는 부시스템(Subsystem)들은 다음과 같은 것이 있다.

— Electronic-power systems
— Stability systems
— Thermal-control systems
— Propulsion systems
— Systems structure.

이러한 시스템들은 목표 시스템 내의 컴퓨터 처리기와 직접 또는 간접적으로 전자적인 측면에서 연계되어 있다.

예상되는 Back-Door Coupling 기술중의 하나는 적절히 제어 되어진(Carefully controlled) 전자장 고주파(Spike)를 이용하여 바이러스를 목표 시스템에 주입시키는 것이다.

또다른 Back-Door Coupling 기술에는 Component Design Tampering이 있다.

Component Design Tampering은 서구국가에서 사용하는 컴퓨터 처리기를 거의 맹목적으로 적대국들이 복사한후 자국의 시스템

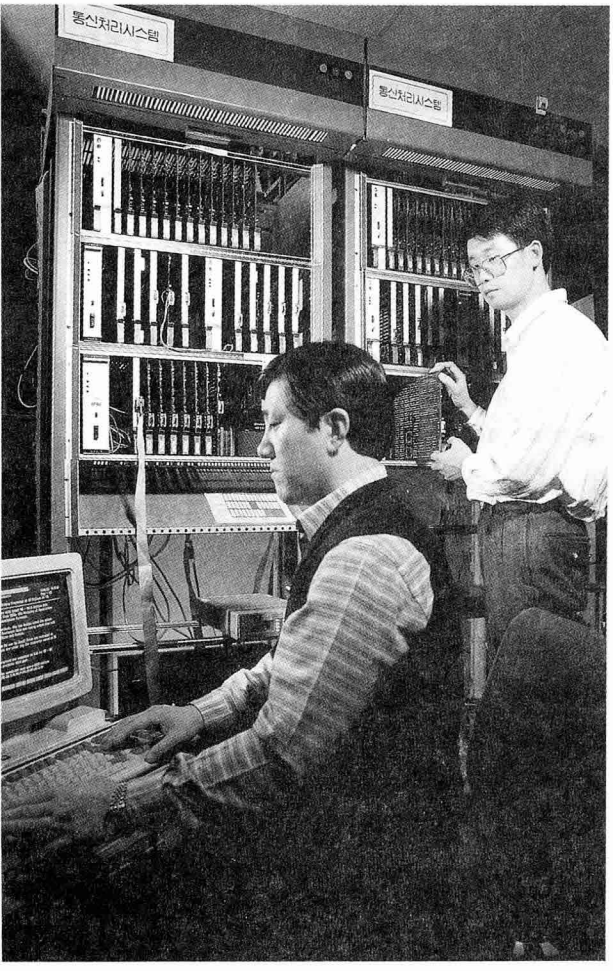
들에 사용하고 있다는 사실을 교묘히 이용할때 사용되어지는 방식이다.

바이러스에 감염된 컴퓨터 처리기의 설계 특성들을 훔쳐서 시스템을 구축함에 따라 상대방은 자신의 시스템내에 바이러스가 들어가 있다는 사실도 모르게 된다.

Front 그리고 Back-Door Coupling 이외에도 바이러스의 전염성을 이용하여 목표 체계와 직접 또는 간접적으로 CVCM 체계가 연계(Coupling) 될 수 있다.

• Direct Coupling

바이러스를 목표 시스템내의 컴퓨터 처리기와 연계시키는 가장 직접적인 방법은 바이러스를 공격 목표 체계내에 직접 주입시키는 것인데, 방법의 성격상 수행하기에 항상 쉽지만은 않다.



이 방법은 공격 목표 체계가 자신에게 오는 전파를 수신하고 있는 동안, 바이러스 프로그램을 계속 전송해 주는 방식이다.

여기서의 의도는 바이러스 프로그램을 상대방에 계속 전송하다 보면, 어느순간 정상적인 전송 자료와 섞여서 바이러스 프로그램이 적의 수신부로 들어가리라는 것이다.

이 방법은 성공만 하면 목표 체계를 확실하게 바이러스에 감염되게 할 수 있다. 이 방법의 단점은 공격목표 대상 자체가 전체 시스템내에서 가장 취약한 통신 Link 부분이 아닐 수 있다는 사실이다.

만약 목표물이 고가의 시스템인 경우라면 외부로부터의 신호연계(Coupling)를 방지하기 위한 고도의 방어 기제를 갖추고 있을 것이다. 이런 경우에는, Indirect 연계가 보다 나은 방법이다.

• Indirect Coupling

CVCM 체계에서 가장 매력적인 방법은 Indirect-Coupling이다. Indirect-Coupling은 바이러스의 전염성질을 이용하는 방식이다.

Indirect-Coupling의 기본 철학은 가장 방어가 안된 부분을 연계(Coupling)후 바이러스를 주입시킬 초기 점으로 취할 수 있으며, 그점에서 바이러스가 의도하는 목표 체계로 전파되리라는 생각이다.

연계 초기점에서 의도하는 목표 체계로 바이러스를 전달하는 또다른 기술은 컴퓨터 체계의 정비 및 진단 도구들을 이용하는 방법이다. 컴퓨터 처리기의 정기 점검은 보통 진단 프로그램을 사용하여 이루어진다.

CVCM 시나리오

CVCM 시나리오 / 전자전임무	TROJAN HORSE	FORCED QUARANTINZ	OVERLOAD	PROBE	ASSASSIN
DENY	○				○
DEGRADE		○		○	
DECEIVE	○				
DELAY			○		
EXPLOIT	○			○	

바이러스에 감염된 처리기(Processor)상에서 진단 프로그램이 실행될때 바이러스가 진단 프로그램으로 전파된다.

이렇게 감염된 진단 프로그램에 의해 진단되어지는 처리기는 모두가 똑같이 감염된다. 이와같은 방식으로 전파되는 바이러스 프로그램은 매우 간단하게 만들 수 있다.

인접한 곳으로의 전파(Propagation)는 CVCM의 독특한 특징이다. 얼마전까지만 해도 방해 체계(Counter-measure system)는 인접한 곳으로 전파를 하지 않았기 때문에, 오늘날의 C^{3}I 체계는 이러한 전파의 문제를 고려하지 않고 설계하였다.

대부분의 전통적인 위협은 Direct Coupling을 통해서 이루어졌기 때문에 오늘날의 C^{3}I 체계는 이러한 위협만을 고려하여 설계하였다.

문제는 C^{3}I 체계내의 아무리 고도로 방어가 되어 있는 중요한 Node라 해도 전파 가능한 방해책(Countermeasure)의 위협에 노출될 수 밖에 없는데, 그 이유는 이러한 C^{3}I의 Node들도 결국에 가서는 외부위협에 대한 대처가 미비한 Node들에 연결될 수 밖에 없기 때문이다.

ECM과 같은 전파되지 않는 방해책의 경우에는, Node에 직접 연결되어 있는 모든 Link들이 안전할 때 그 Node는 안전하다고 하는데, CVCM과 같이 전파되는 방해책의 경우에는, Node에 직접 또는 간접적으로 연결되는 모든 Link들이 안전할 때, 그 Node는 안전하다고 한다.

CVCM과 같이 전파되는 방해책의 출현으로 적의 통신 네트워크를 이용하여 상대 시스템을 공격할 수 있는 방법이 생겼는데, 이 방법은 우리측 체계의 안전성에도 새로운 위협의 대상이 되고 있다.

CVCM의 Scenarios

CVCM 체계에서 바이러스를 전개할때 사용할 수 있는 작전 시나리오는 여러개가 있는데, 그중 몇개를 소개하면 다음과 같다.

• Trojan Horse Scenario

Trojan Horse Scenario는 그 이름이 의미하는 바와 같다. 목표 체계에 투입된 바이러스는 미리 정해진 시간이나 사건(Event)이 발생할 때까지 잠복해 있다가 시스템에 결정적인 피해를 주게 된다.

이 시나리오의 장점은 원하는 사건(Event)이 일어날 때까지는 효과를 나타내지 않기 때문에 상대방의 의혹을 살 여지가 없다는 사실이다.

• Forced Quarantine Scenario

상대방의 컴퓨터 네트워크를 공격할때, Forced Quarantine Scenario가 이용되어진다. 네트워크에 들어간 이 바이러스는 자신의 존재를 노출시키게 된다.

네트워크의 다른 Node들 또한 감염될지 모른다는 우려에서 감염된 노드는 외부와 차단 되게 된다. 따라서 네트워크의 효능이 크게 감소하게 된다.

• Overload Scenario

이 바이러스는 시스템내에서 자기 복제(Duplicate)를 여러번하여 시스템의 처리속도를 늦추게 한다.

이러한 시스템의 처리속도 지연은 Fire-Control Radar와 같이 시간에 민감한 시스템에 있어서는 치명적인 사항이다.

• Probe Scenario

이 바이러스는 특정 자료의 위치를 찾고자 할때 사용된다. 일단 찾고자하는 자료를 확인한 후 이 바이러스는 특정 위치로 되돌아오게된다.

이러한 성질을 이용하여 매우 중요시 되는 정보를 찾아 이용할 수 있게 된다.

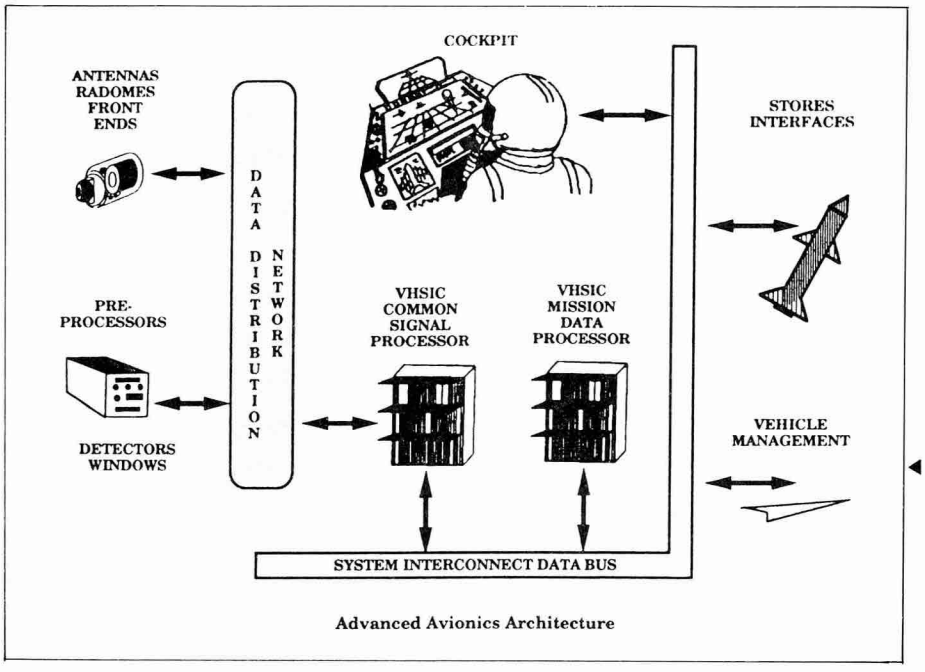
• Assassin Scenario

네트워크 내의 특정 파일이나 시스템을 파괴하고자 할때 이 바이러스가 이용된다. 네트워크 내에 삽입된 이 바이러스는 흔적을 남기지 않으면서 목표물을 찾아 이동하게 된다.

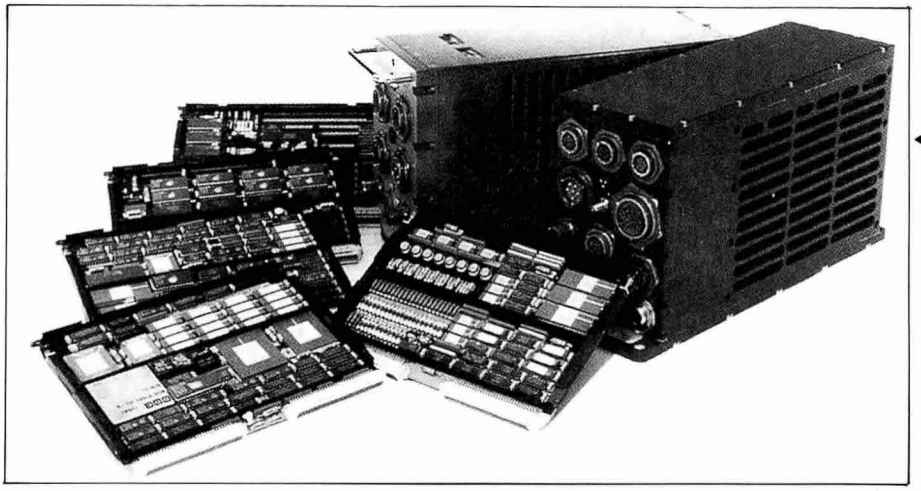
목표물에 도달하면 이 바이러스는 목표물의 기능을 마비시키며, 동시에 자신을 지워버린다(Erase). 따라서 어떠한 흔적도 남기지 않게 된다.

시스템의 例(System Example)

앞으로는 아래와 같은 시스템 경향때문에 CVCM이 가능해진다.



◀ 모듈화된 단일 항전 체계와, 고속의 Data Bus을 이용한, 기능의 통합이라는 개념에 바탕을 두고 오늘날의 첨단 항공전자 체계가 발전하고 있다



◀바이러스를 목표 시스템내의 컴퓨터 처리기와 연계시키는 가장 직접적인 방법은 바이러스를 공격 목표 체계내에 직접 주입시키는 것인데, 이 방법은 성공만 하면 목표 체계를 확실하게 바이러스에 감염되게 할 수 있다

• 첨단의 항전 구조

(Advanced Avionics Architectures)

오늘날 항공전자 계통 구조(Avionics Architectures)의 설계가 디지탈, 기능별 통합(Functional Integration) 그리고 공통화(Commonality)를 보다 많이 사용하게 되면서 첨단의 항전 계통 구조가 가능케 되었다.

베트남 전쟁 당시의 항전계통은 여러개의 디지탈 또는 아날로그 Subsystem들로 되어 있는데, 개개의 Subsystem들은 각각 Line-Replaceable Unit들이었다. 오늘날의 장비들은 디지탈 형태의 Subsystem들로 되어있는데, 아직도 Line-Replaceable Unit형태를 유지하고 있으나, 개개의 Subsystem들은 고속의 멀티플랙스 형태의 데이타 버스를 이용하여 상호 연결되어 있다.

향후의 발전된 항전계통 구조는 단일의 항전 Subsystem으로 되는데, 고속의 데이타 버스를 이용하여 통합된 기능(Integrated Function)과 모듈단위의 Packaging이 가능하게 될것이다.

• 통합된 지휘 통제 체제(Integrated Command and Control Systems)

발전된 무기체계들에서 시기 적절한 자료의 수요가 증가함에 따라, 이와같은 자료를 Tactical Joint와 합동군(Unified Forces)과 같은 여러 사용자들에게 전송하고 제공하기 위해 복잡한 통신 네트워크가 개발되어 오고 있다.

美 육군은 기동통제(Maneuver Control), 화력지원(Fire Support), 방공(Air Defence), 정보 및 전자전(Intelligence/Electronic Warfare), 전투지원들간의 통신을 통합할 계획을 하고 있다.

이러한 계획은 발전된 기능별 연결을 가능케할 통신수단의 현대화가 이루어질때 가능한 것이다. 이러한 통신 계획 핵심은 시중에서 구입 가능한 상용의 공통(Common) 소프트웨어와 하드웨어를 통신체계에 사용하는 것이다.

대비책(Protection)

CVCM을 이용하여 적국의 여러 목표물을 공격할 수 있다. 상대방의 CVCM공격에 대한 효율적인 대비책을 고안하면서 요구되는 사

항, 전략 그리고 가용한 기술에 대한 이해가 있어야 한다.

• 요구되는 사항들(Requirements)

CVCM의 공격에 대한 대비책은 전통적인 ECM공격에 대한 대비책에서 볼 수 없는 미묘한 문제가 따른다.

적대행위가 진행될때 컴퓨터 네트워크의 중요한 Node에 대한 공격에 대비하기 위해 중요한 Node와 직접적으로 연결되어 있는 모든 통신 Link들은 CVCM뿐만 아니라 전통적인 ECM에 대해서도 대비되어 있어야 한다.

더욱이 CVCM에 대한 대비를 위해서 중요한 Node가 들어있는 컴퓨터 네트워크내의 모든 Link들이 보호되어져야만 하는데 CVCM은 공통 Link를 통해서 한 Node에서 다른 Node로 전파(Propogate)되기 때문이다.

이외에도 ECM에 대항해서 보통 사용되어지는 Multinode 방어책(Protection Scheme)은 CVCM에 대해서 별다른 효과가 없다.

그 이유는 평상시 적의 시스템에 이식된 CVCM의 효과가 교전시까지도 지속되기 때문이다.

중요한 Node가 들어있는 네트워크내의 모든 Link는 전시와 평상시에 구분없이 항상 보호되어야 한다.

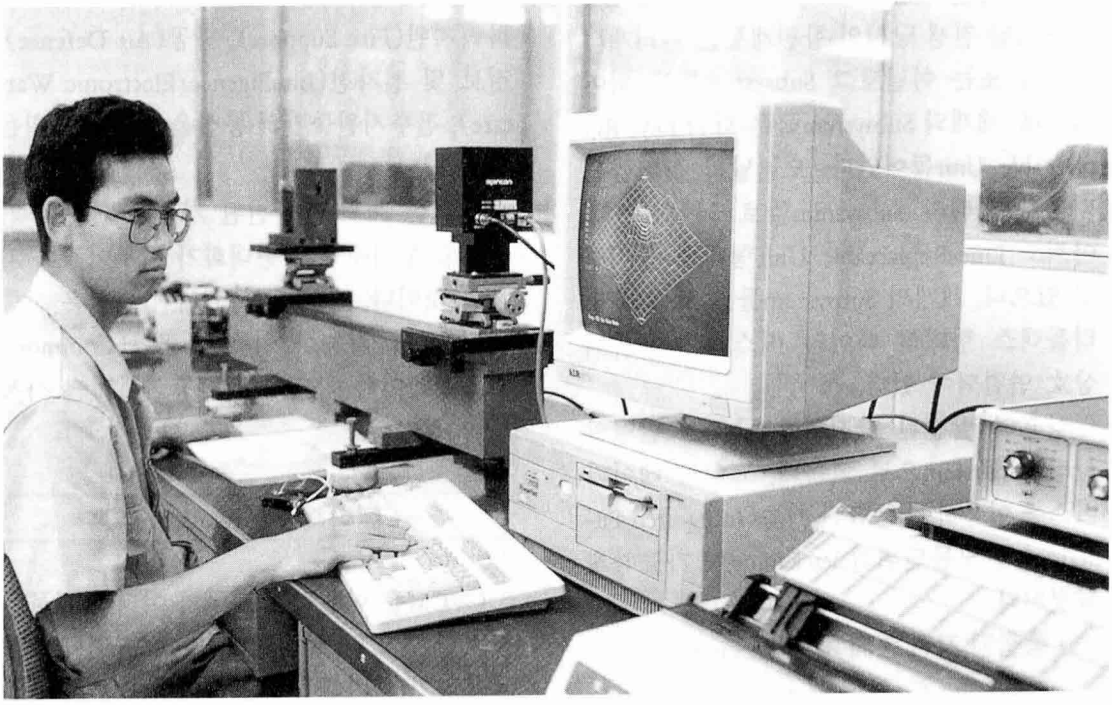
• 전 략(Strategy)

효율적인 CVCM 방어 전략에는 여러 방어 단계가 있다.

* Level I – 접근 거부(Deny Access)

방어의 첫번째 단계로서 외부로 부터 침투 목적의 소프트웨어가 시스템내에 들어오지 못하게 하는 것이다.

* Level II – 발견(Detect)



외부로부터 CVCM이 시스템내에 항상 들어오지 못하도록 할 수 있는것이 아니라는 인식하에서, 다음단계는 시스템의 바이러스 감염여부를 발견하는 것이다.

* Level Ⅲ－봉쇄(Contain)

바이러스 프로그램의 중요성질은 감염된 시스템내에서 확산(Propagate) 할 수 있다는 사실이다.

따라서 감염된 부분을 외부와 차단하는 봉쇄 방안을 통해서 이러한 확산을 막는 것이 중요하다.

* Level Ⅳ－제거(Eradicate)

바이러스 프로그램이 궁극적으로 외부의 방어책을 뚫고 시스템으로 침투하리라는 가정하에서 큰 피해가 발생하기 이전에 바이러스 코드를 제거하는 것이 무엇보다도 중요하다.

* Level Ⅴ－회복(Recover)

바이러스 프로그램이 제거되기전에 데이타 파일이나 프로그램이 들어있는 파일에 커다란 피해를 입힌 경우에는 최신의 Back-up 파일을 이용하여 시스템의 회복을 꾀하는 것이 현명한 방안이다.

* Level Ⅵ－대체 시스템의 준비(Provide Alternative Operation)

바이러스에 대한 기술적인 해결 방안이 불가능하거나, 때에 맞추어 가능하지 않은 경우가 있다.

고도의 정교한 바이러스 프로그램이 예기치 못한 상태에서 침투할 때 이러한 상황이 발생할 수 있다.

이러한 가능성에 대한 대비책으로 바이러스에 의해 특정 시스템이 무력화되어 있는 상황에서도 작전할 수 있는 대체 계획을 준비해야 한다.

방어 기술(Protection Techniques)

CVCM에 대항한, 효율적인 방어 전략을 구현하기 위해서는 효율적인 하드웨어 및 소프트웨어 전략 그리고 숙련된 운영의 묘가 요구되어진다.

• 하드웨어(Hardware)

* 외부로부터 실행 소프트웨어 프로그램으로의 접근을 막기위해서는 프로그램 가능한 Read-Only Memory(PROM)나 컴펙트 디스크 또는 다른 Read-Only Memory를 사용하시오.

* 전파되는(Spreading) 바이러스를 봉쇄하기 위해서 시스템을 전기적(Electrically)으로 외부와 차단하시오.

* 바이러스의 전파 매체(Media)를 없애기 위해 여러 종류의 다중 프로세서들을 통합(Integrate)시키시오.

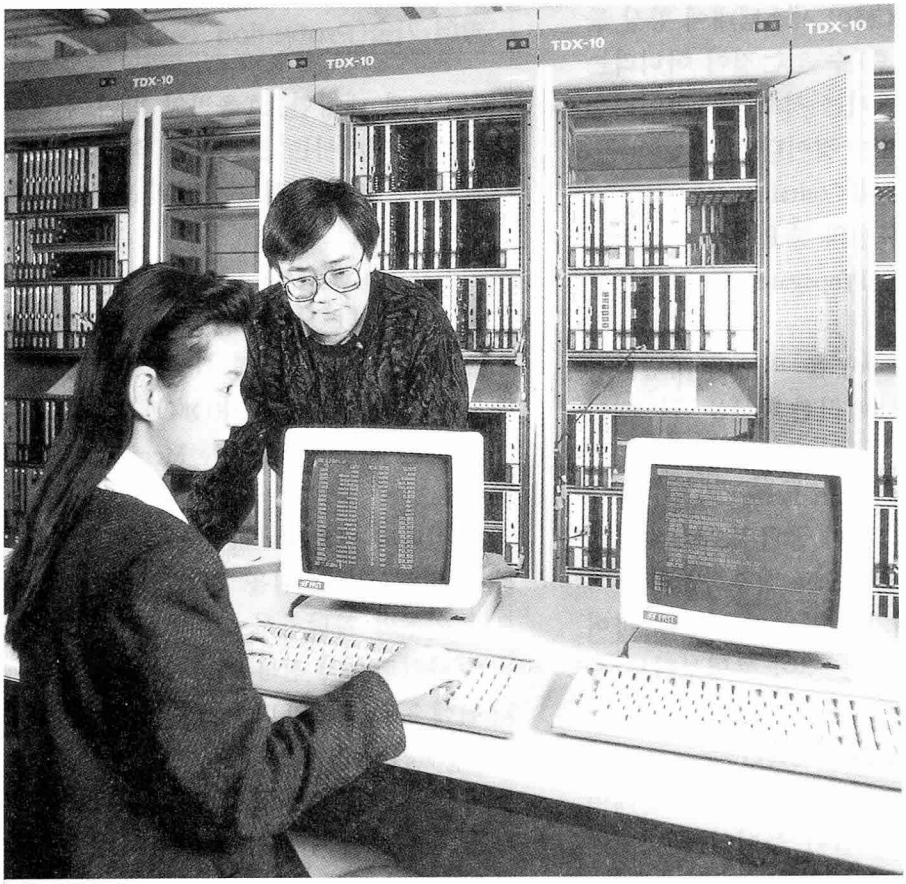
이러한 통합된 시스템에서 바이러스가 효과를 발휘하기 위해서는 하나의 바이러스 프로그램이 개개의 마이크로 프로세서의 명령 체계(Instruction Set)내에서 동시에 작동할 수 있어야 한다.

• 소프트웨어(Software)

* 중앙 처리장치(Central Processing Unit) 근처에 허가되지 않는 기능을 수행하는 프로그램들이 접근하지 못하도록 하시오.

* 면역 프로그램(Immunization Program)을 이용하여 시스템 프로그램내에 소프트웨어 방어책(Protection)을 형성하여, 바이러스 소프트웨어가 내부에 침입하여 감염시키는 행위를 하지 못하도록 하시오.

* 바이러스의 침입 유무를 감지하기 위해서 보유하고 있는 소프트웨어에 대한 지속적인 관찰을 게을리하지 마시오.



◀바이러스는 수백만줄의 소프트웨어내에 잠복해 있다가 결정적인 순간에 출현할 수 있으므로, CVCM에 대한 방어책은 적이 이러한 바이러스를 이용하기 전에 실시 되어야 한다

* 바이러스의 전파를 통제하기 위하여 여러개의 운영 체계(Operating Systems)를 통합(Integrate)하시오.

이러한 환경하에서 효과를 발휘하기 위해서 바이러스 프로그램은 개개의 시스템에서 사용되는 언어들을 동시에 말할 수 있어야 한다.

* 바이러스 제거 프로그램을 이용하여 침입한 바이러스를 제거하시오.

이러한 제거 프로그램들은 감염된 소프트웨어를 처리한후, 바이러스 프로그램을 감염된 소프트웨어로 부터 분리해 준다.

* 바이러스로 부터의 복구책으로 보관하고있는 소프트웨어를 다시 Load 시키시오.

이러한 방법에서는 감염된 프로그램이나 파일들을 지워버린후, 감염이 안된 복사본을 이용 다시, 소프트웨어를 설치하게 된다.

감염된 프로그램으로 부터 바이러스를 제거하기 보다는, 이러한 방법이 보다 손쉬운 경우가 종종 있다.

• 운영(Operations)

* 바이러스 프로그램에의 노출을 최소화하고, 중요한 시스템 프로그램에의 바이러스 접근을 제한 하기위해서 조직화된 안보 대책을 강구하시오.

이러한 안보 대책의 일환으로 바이러스의 침투 예상 경로를 차단 할 수도 있다.

* 바이러스를 탐지하기 위해 시스템의

활동을 관찰하시오.

아무리 좋은 방비책도 시기 적절하게, 그리고 일관성있게 사용되지 못한다면 효과가 없다. 가장 좋은 방비책은 바이러스의 시기 적절한 발견이다.

* 바이러스의 전파를 억제하기 위하여 엄격한 운영 보안을 실시토록 하시오.

* 바이러스의 전파를 억제하기 위하여 바이러스에 감염된 소프트웨어를 갖고 있는 사용자들의 접근을 제한하시오.

* 바이러스가 치명적인 피해를 야기시키는 경우에 대비하여 비상 대책을 강구하시오.

임의의 응용소프트웨어에 대한 방어책을 설계하면서, 그 소프트웨어를 손실했다고 가상했을때의 비용과 방어책의 강구에 드는 비용을 사려깊게 비교해 보아야 한다.

개념 설정 단계에 있는 시스템의 보호 방안에 대한 제한은 없으나, 야전에 배치된 시스템을 위해 취할 수 있는 유일한 가용 방어책은 운영 절차(Operational Procedure)임을 명심해야 한다.

오늘날의 새로운 군사 전자 시스템 기술은 여러 각도에서 유익한 점이 많이 있지만, 불행스럽게도 컴퓨터 바이러스에 의한 시스템의 침투 가능성을 높여 놓았다.

그러므로 향후 상업용 통신과 컴퓨터 시스템 기술을 설계 및 운영 할때는 이러한 바이러스들의 치명적인 능력들을 고려해야 한다.

방해책 기술에 대항하여 사용되고 있는 종전의 방어 전략들이 CVCM에 크게 효과가 있을 수 있다.

바이러스는 수백만 줄의 소프트웨어내에 잠복해 있다가 결정적인 순간에 출현할 수 있으므로, CVCM에 대한 방어책은 적이 이러한 바이러스를 이용하기 전에 실시 되어야 한다. *

참 고 자 료

▲Dr. Myron L. Cramer and Stephen R. Pratt, 「Computer Virus Computermeasures A New Type of EW」, 〈Defence Electronics〉, pp.75~84, 1989년 10월호

▲이민규, 「효과적 군 홍보를 위한 뉴미디어 활용방법」, 국방일보 현역 기자 교육용 참고자료

▲James W. Rawles, 「The Viral Threat」, 〈Defence Electronics〉, pp.62~67, 1990년 2월호

▲John Paul Newport Jr, 「A Growing Gap In Software」, 〈Fortune〉, pp.132~142, 1986년 4월 28일