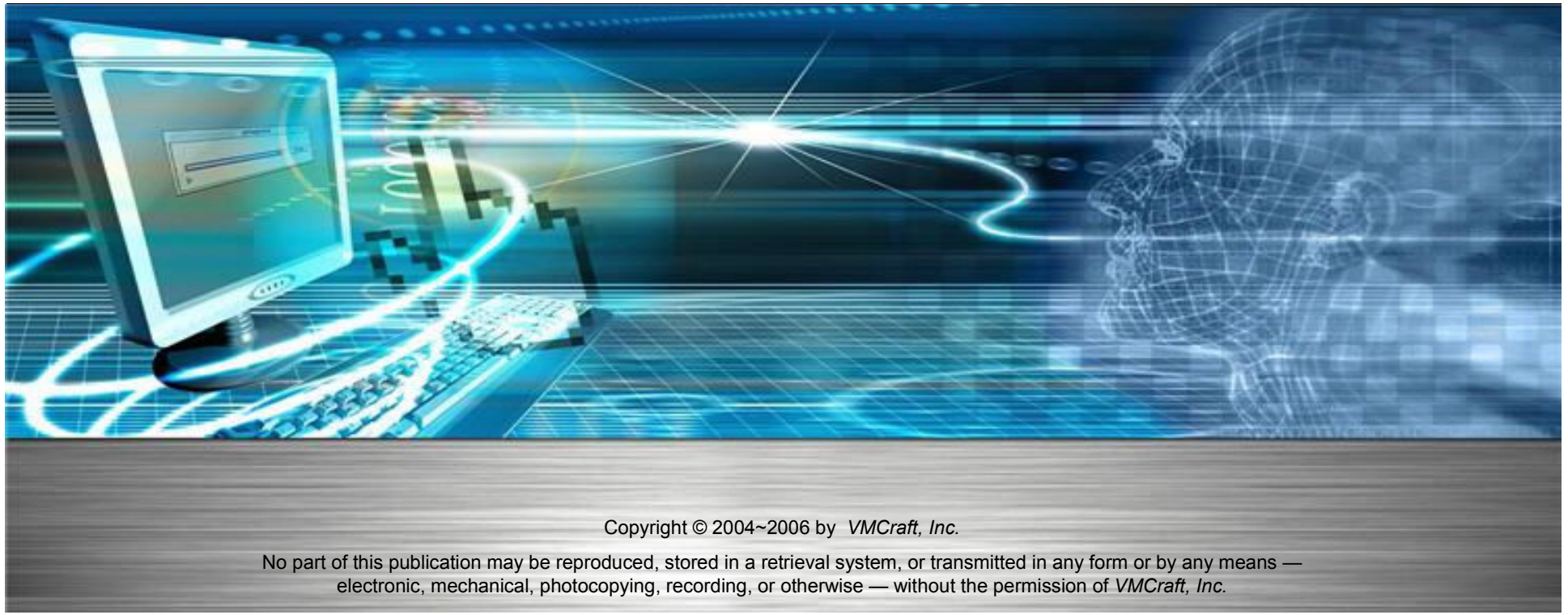
VMCraft, Inc.

www.vmcraft.com

# Virtualization Architecture

# for the Security of Web and Application

**NOV. 16. 2006**



# I. 어플리케이션 보안 현황



- Key Points of 2006
- Application Vulnerability?
- High incidence of crime
- Fire warning shot

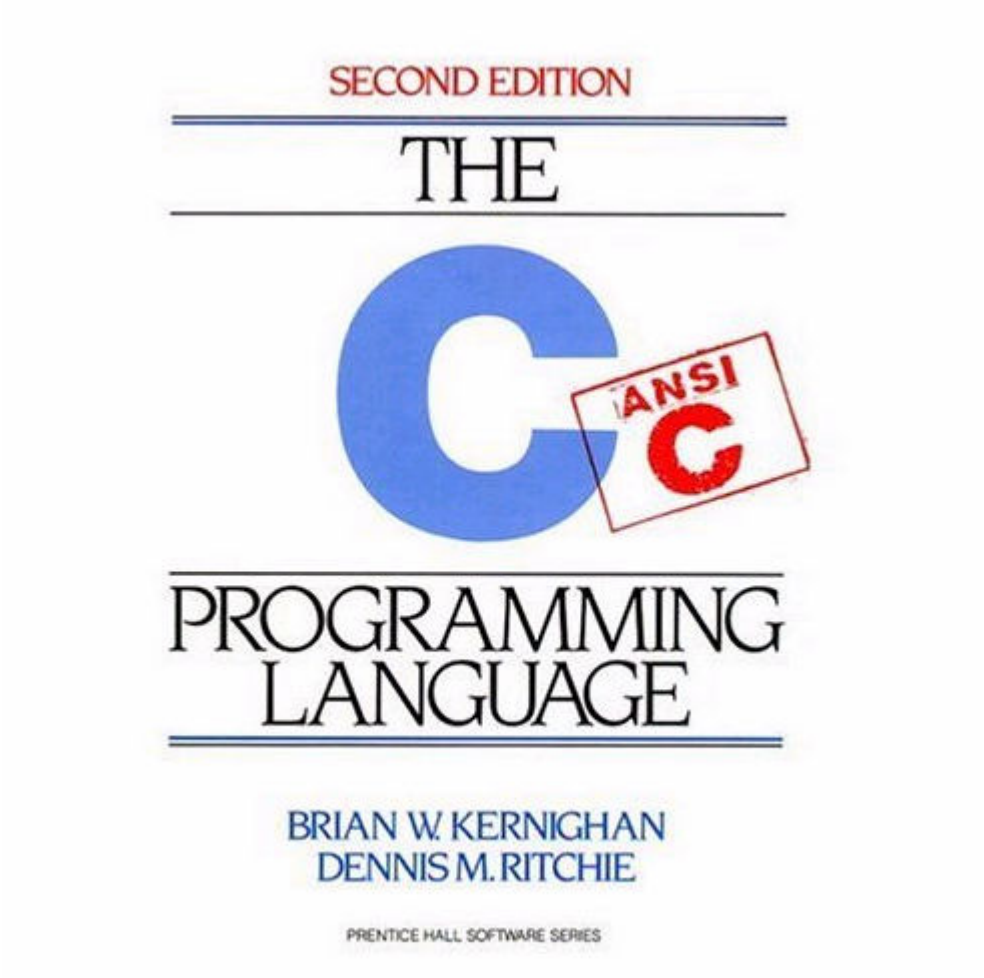
# Key Points of 2006

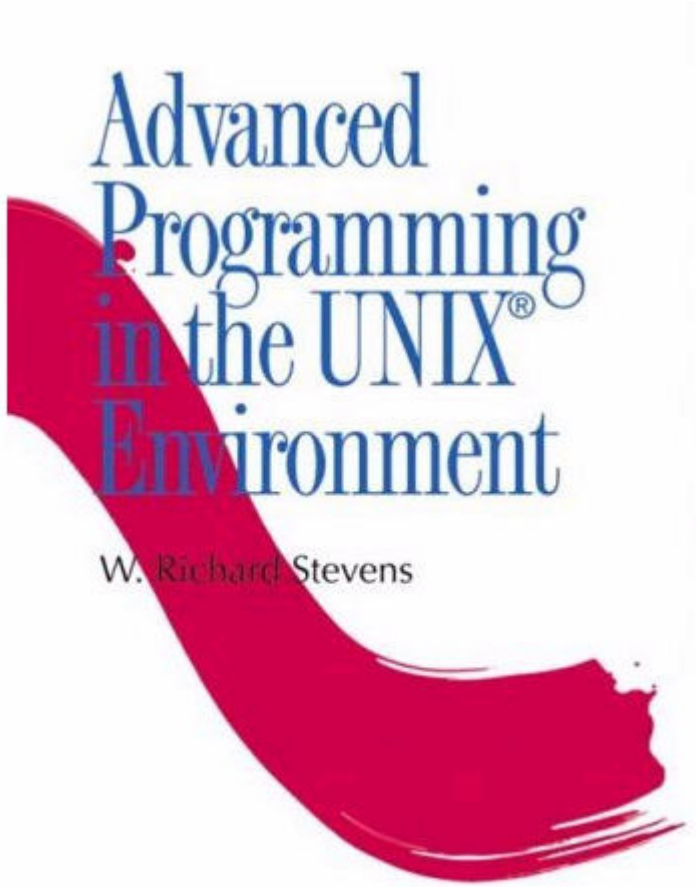


- Our working environment is changing rap?
- Most attacks are based on web or application vulnerabilities
- Web Firewalls no longer provide complete safety
- Virus, Worm and hacking techniques are merging
  - Permissions escalations
  - Data theft
  - Automation
- Common methods of an attacking application
  - Reverse Engineering(Static Analysis), API Hooking(Dynamic Analysis)
  - Fuzzing(Dynamic Analysis)
  - Google Code Search

# Application Vulnerability? #1





# Application Vulnerability #2



어플리케이션 취약점은 개발자가 미처 생각지 못했던 에러 및 각종 우회 기법 등을 이용해 개발자가 의도하지 않았던 각종 전자적 침해 요소를 통한 다양한 공격이 가능

| Flow Of Attack | |
|---|---|
| Release | ActiveX Control, Download |
| Business Process Analysis | 인증 권한은 '1 < X < 3'의 조건을 만족해야 권한 실행이 가능 |
| Developer | "정상적인 관리자는 '2'를 입력함으로써 인증에 성공할 것이다" |
| Attacker | '1.5'를 입력 함으로써 개발자의 의도를 우회 함.I |
| Response | '1.5'를 필터링 함으로써 문제를 해결 함.<br>그렇다면 '1.1', '1.12' 등의 입력 대책은??? |

# High incidence of crime




[Case #1]
IGE에 따르면 게임아이템의 연간거래규모는 2005년 15억 달러였으며, 2006년에는 약 27억 달러에 육박할 것으로 예상.

"최근 5년간 한국에서 발생한 사이버범죄의 40%가 게임관련 범죄이며, 게임범죄와 관련된 고소 및 고발 사건이 전체 사건의 21.3%을 차지함"

Source : KISA(Korea Information Security Agency)

[Case #2]
Internet Banking : 고객수 2,290만명, 거래의 34%
Cyber Securities Financing : 전체 거래의 60%

Source : BoK(Bank of Korea) 2005. 07

Source : wired.com – IGE (Internet Game Entertainment)에서 유통되는 게임 아이템 거래 가격

# Fire warning shot #1



**Case 1**

- 특정 기법 최초 보고 후 3~5년 후 대중적으로 이슈화 됨
- 대중적인 공격기법이 되는 시점은 script kid 용 자동화 툴 발표 시점과 연관됨
- 2007년 이후 대중적으로 이슈화될 공격기법들은 기존 보안솔루션으론 방어하기 힘든 부분이 있음

### 사례1: Game Hacking

- 1999년경 ActiveX 취약점 형태 최초 보고
- 2001년경 고급 해커들 사이에서 이슈화
- 2005년 중국발 해킹사고 빈번하게 발생
- 2006년 한해 6000건 돌파 예상

### 사례2: Web Proxy

- 2000년경 Achilles 발표
- 2001년경 고급 해커들 사이에서 이슈화
- 2005년 민원서류 위·변조 이슈화
- 2006년 현재 웹 서버 해킹 시 대부분 web proxy 이용

# Fire warning shot #2



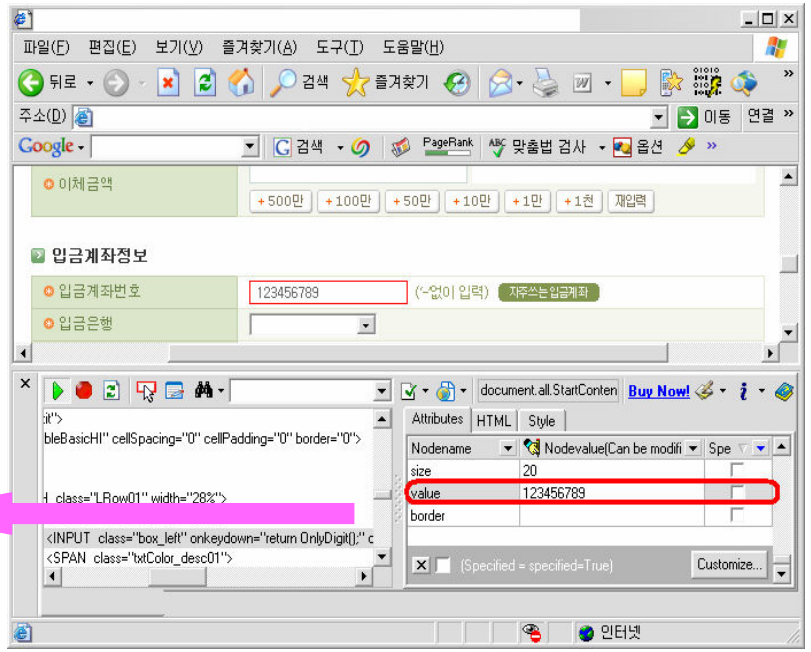
**Case 2**

- 인터넷 뱅킹 시스템 공격용 웜 / 트로이목마 등장 가능성
- 특정 일에 발생하는 모든 계좌이체의 입금계좌번호를 조작
- 계좌이체 비밀번호 등 중요정보를 공격자에게 전송
- One time password로는 해결되지 않는 문제점

## 사례3: 웜/트로이목마 시나리오

- DOM 구조 상에서 입금계좌번호 조작
- 불특정 다수에게 송금되도록 조작할 경우 큰 혼란
- 검색엔진의 검색결과를 조작하는 형태의 애드웨어는 이미 배포되고 있음

이체직전 계좌번호를 임의로 수정할 수 있으며, 이체 결과페이지도 조작할 수 있다.



# Fire warning shot #3



**Case 3**

- 리버스엔지니어링을 이용해 주요 로직의 runtime fabrication 가능
- 발견된 기법에 대한 자동화 공격 툴 제작

## 사례4: 어플리케이션 분석

- 인터넷 뱅킹용 어플리케이션 리버스엔니어링.
- 내부로직 분석

출처: 국내 모 보안연구그룹

```
seg000:16002000          dd 0BF571F5h, 33AD3970h, 858E43CBh, 0D44BE1F5h, 0CBAB9AC3h
seg000:16002000          dd 474B43DEh, 0EB4558EAh, 0F2544091h, 4C5641B5h, 0D966F149h
seg000:16002000          dd 0FDA1704Bh, 0CFE2990Ch, 68C60C8Bh, 3A8EEF96h, 38E648C6h
seg000:16002000          dd 9AC24941h, 93CBA119h, 0E2F7E2A7h, 9755D120h, 0CA74F32Fh
seg000:16002000          dd 0BCE87DF5h, 2A3B205Fh
seg000:16002800          dd 68E1BDBDh, 1A236CB6h, 0F956458Ch, 32A21DE9h, 0CE7F727Ch
seg000:16002800          dd 0B1E5EFF0h, 1F582FB6h, 7EFF967Ah, 194B5F09h, 769E9A59h
seg000:16002800          dd 0DF7EDD51h, 9FC756D3h, 15027A5Ah, 57DA5DBAh, 6B335283h
seg000:16002800          dd 91D99CEh, 0ABBEBE59h, 655986B8h, 0E770C058h, 3FD73C31h
seg000:16002800          dd 94217B21h, 27B0EB73h, 71D492A9h, 0E96727B0h, 7B76BA12h
seg000:16002800          dd 0EA6F0ACBh, 37F820D3h, 0F41512F6h, 0B6A7C1E3h, 822D89DFh
seg000:16002800          dd 7D9144FDh, 286F42D0h, 0C473958Eh, 0FC94CBDAh, 6F9B6A26h
seg000:16002800          dd 0ABC9FB0Dh, 2BC1C7B6h, 3020A614h, 0C5D12913h, 7F0EA9D5h
seg000:16002800          dd 10B79EF9h, 7A7CE961h, 331C46D6h, 4B99EF4Dh, 94DBE674h
seg000:16002800          dd 0C45BF6AAh, 0BB36BF73h, 0B5FE86D6h, 75DCF655h, 9F0D56B6h
seg000:16002800          dd 0B69F46A0h, 35825A9Ch, 5D104C61h, 539166A8h, 7C768821h
seg000:16002800          dd 64BFE26Ch, 9247A956h, 0ACD2EC8h, 0E0752B2Eh, 5477F389h
seg000:16002800          dd 0E83F0A7Eh, 1A9E88A6h, 9AAFF0E5h, 0ABE82318h, 0DF2EB168h
seg000:16002800          dd 1498FBA7h, 614A4A96h, 20EF37BAh, 3AC51FFh, 0E3A5BE8h
seg000:16002800          dd 52706ECCh, 0CCBFF4B9h, 2540C37Dh, 0C9498F3h, 0F8F142Fh
seg000:16002800          dd 0F66586C2h, 7DF354B6h, 31C0B9E7h, 9712413Dh, 23D47E12h
seg000:16002800          dd 0E775FD3h, 0AF8155C9h, 5D319E24h, 2B82413h, 788B76B0h
seg000:16002800          dd 69768E1Eh, 265EF23Ch, 0E112861Fh, 23B548C1h, 0A0E9A738h
seg000:16002800          dd 3ED1FC0Dh, 121E37F6h, 0DBF2708Fh, 0AE9A780Ah, 0E84D4B6Ch
seg000:16002800          dd 0E32FA2D9h, 0DE56ED2Bh, 6CA7CA4Bh, 79EA4063h, 0BE6B8E69h
seg000:16002800          dd 0D957DA87h, 9F489D08h, 5B57DCECh, 6B723B26h, 3C7B4503h
seg000:16002800          dd 0EF1BA76Eh, 46DFFBEh, 0A83D7D2Fh, 0C2AB0D72h, 5EB29925h
seg000:16002800          dd 3E63579Dh, 6D7AE0F4h, 0D76D928Ah, 0E7C31DA6h, 134D79Eh
seg000:16002800          dd 0CC88804Ch, 0DD6865EAh, 1660BB7Bh, 313B538Ch, 0B68381CEh
```

# Fire warning shot #4



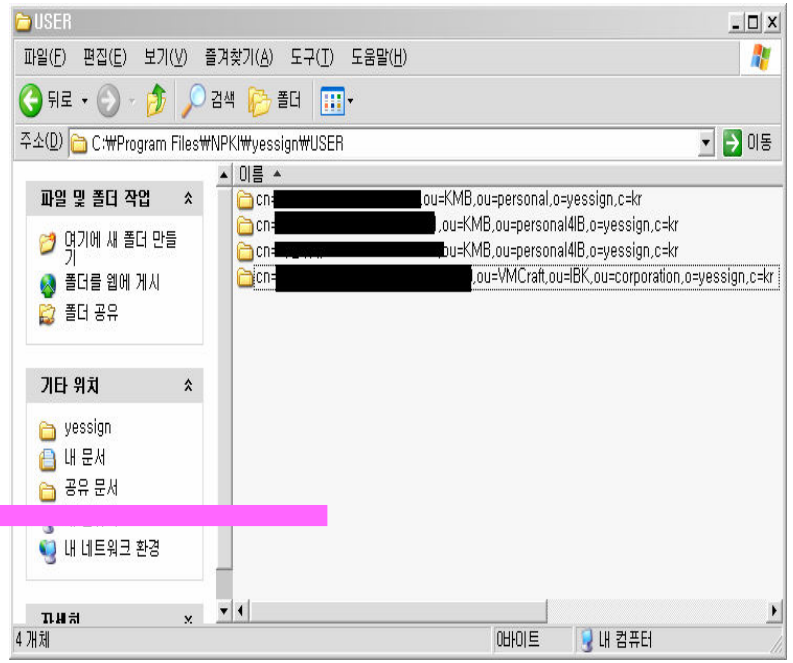
**Case 4**

- 인증서 파일 등 어플리케이션 중요데이터가 유출될 수 있다.
- PC 방 등 공용PC 에서 인터넷 뱅킹을 시도한 경우
- 개인 PC가 해킹된 경우

## 사례5: 인증서 파일 유출

- 공인인증서 파일은 하드디스크 상에 존재한다.
- 암호화된 인증서를 복사하고, 다른 취약점과 결합해 암호를 알아냄
- 타인의 인증서를 도용

**C:\program files\NPKI\yessign\user**
폴더에 인증서 파일 존재



# Fire warning shot #5

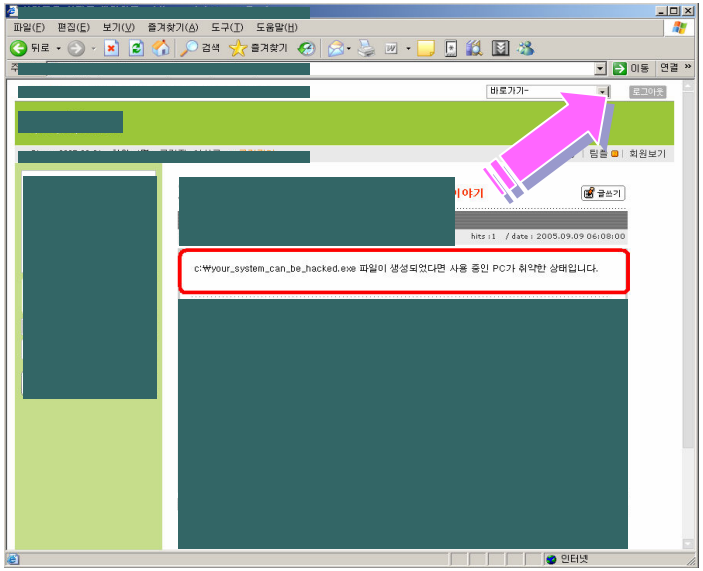


**Case 5**

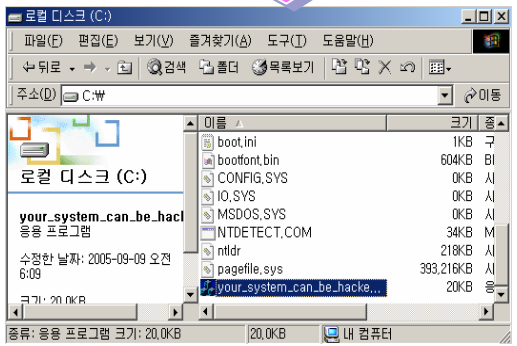
- ActiveX 컨트롤 취약점

- 내부 직원 PC에는 인터넷 뱅킹, 포털 사이트에서 설치한 각종 ActiveX 컨트롤 존재
- 이들 ActiveX 컨트롤에 취약점이 존재할 경우 외부 해커에 의해 PC 해킹 가능
- 테스트 결과 1000만명 이상이 사용하는 ActiveX 컨트롤에 다수의 취약점이 존재함
- 이론적으로 1000만대의 PC의 권한 획득이 가능

숨겨진 PoC 코드 실행




```
<OBJECT classid='CLSID:XXXXX-X' …
<PARAM NAME="url" VALUE="http://">
<PARAM NAME="UpdatePath" VALUcom">
<PARAM NAME="lib" VALUE="hacke3">
</OBJECT>
… 이하생략 …
```



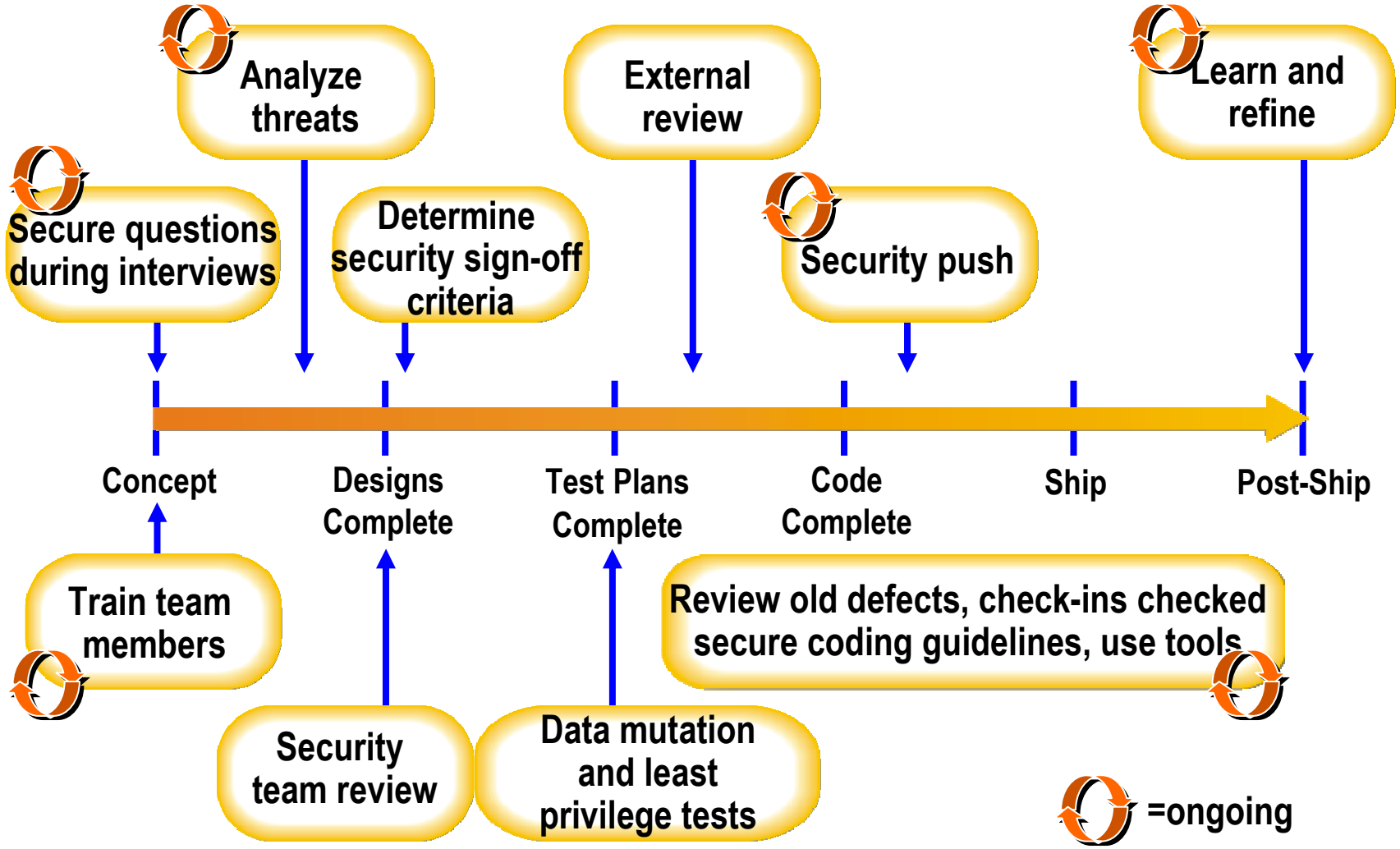
# II. 방어 기술의 한계



- Secure Coding
- Firewall
- Web Firewall
- Anti-Reverse Engineering
- SBC(Server based Computing)

# Secure Coding #1

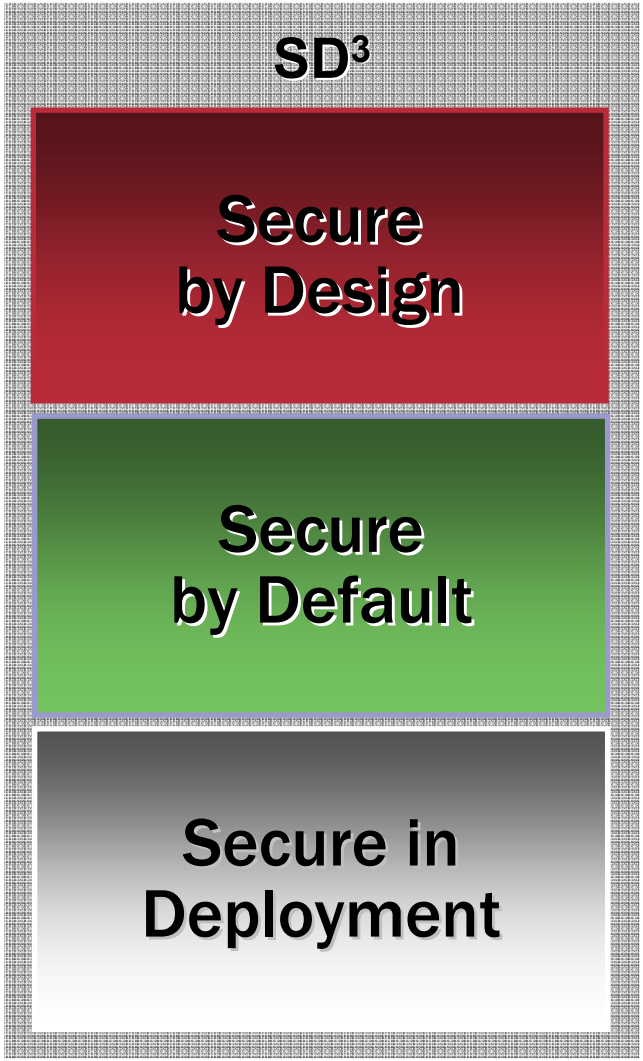





Source : Essentials of Application Security - Microsoft

# Secure Coding #2






- Secure architecture and code
- Threat analysis
- Vulnerability reduction

- Attack surface area reduced
- Unused features turned off by default
- Minimized privileges used

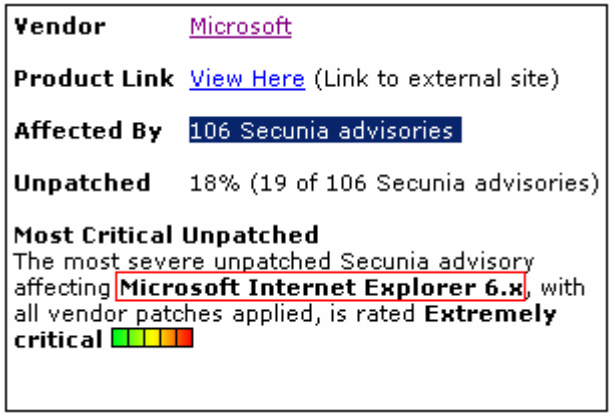
- Protection: Detection, defense, recovery, management
- Process: “How to” guides, architecture guides
- People: Training

Source : SD3 Security Framework - Microsoft

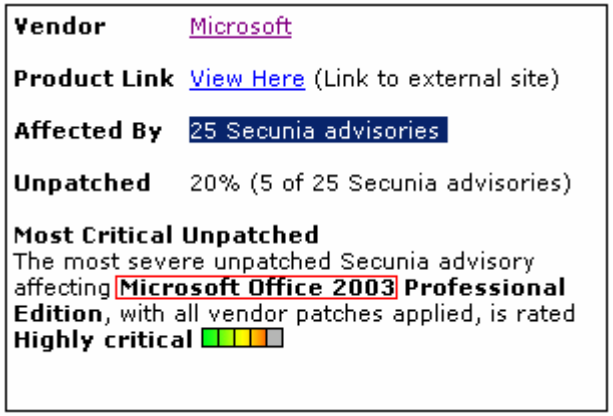
# Secure Coding #3



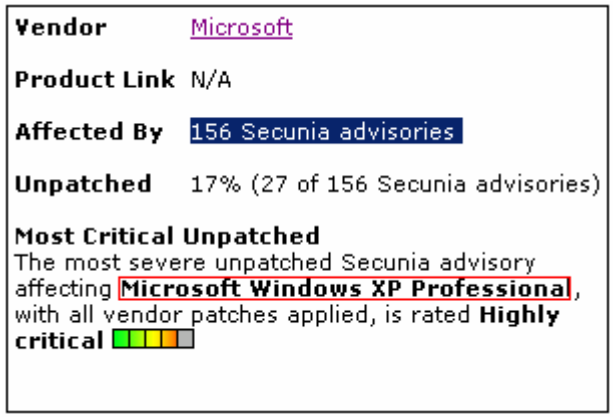
Secure development is a term largely associated with the process of producing reliable, stable, bug-free and vulnerability-free software. There are a number of ways that this can be archived. However, this goal is unlikely.



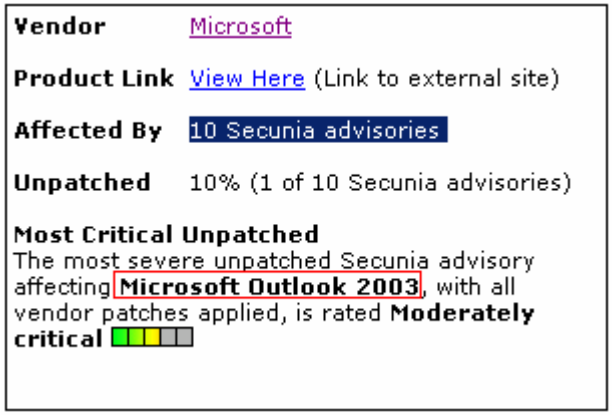

**Internet Explorer 6.x**




**Office 2003**




**Windows XP Professional**




**Outlook 2003**

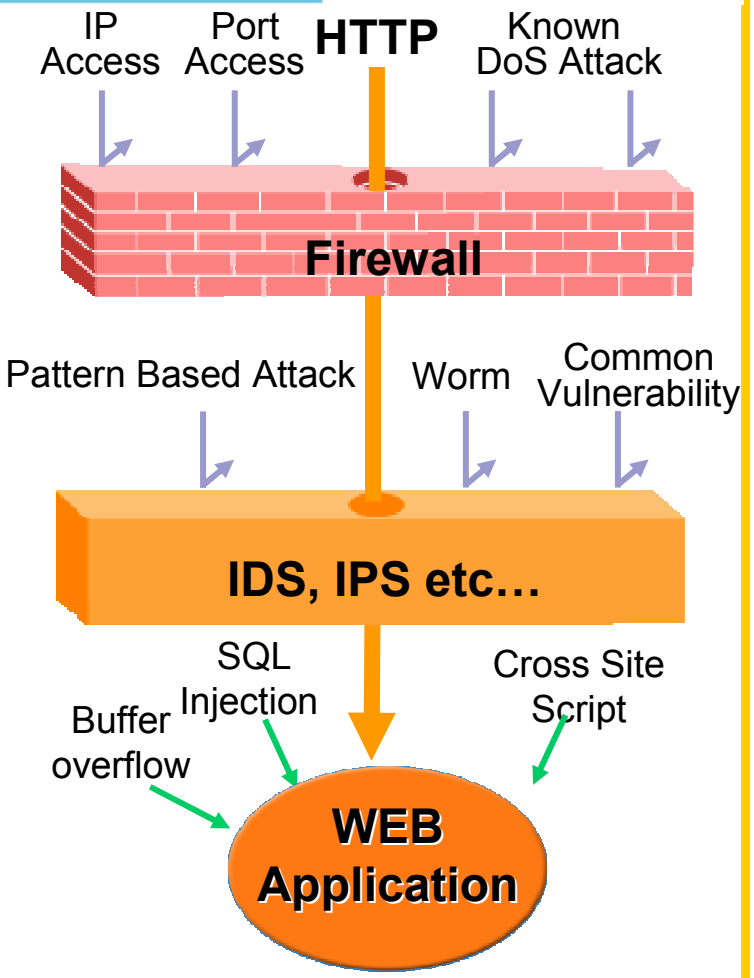
**Source : secunia.com (OCT. 2006.)**

# Firewall



일반적으로 웹 기반의 침해사고가 발생하는 이유는 웹 응용프로그램의 설계시에 데이터베이스 접근, 관리자 인증 및 사용자 인증, 웹 응용프로그램의 설계 방식에 대한 보안성 검토가 이루어지지 않기 때문에 발생합니다.

결론적으로 최근 웹 애플리케이션의 취약점을 이용한 공격이 폭발적으로 증가하는 이유는 웹 애플리케이션에 존재하는 취약점의 보안성 검토를 수행하지 못 하기 때문이며, 웹 애플리케이션 취약점이 존재할 경우, 현재의 보안 아키텍처 구성으로는 방어가 불가능합니다.



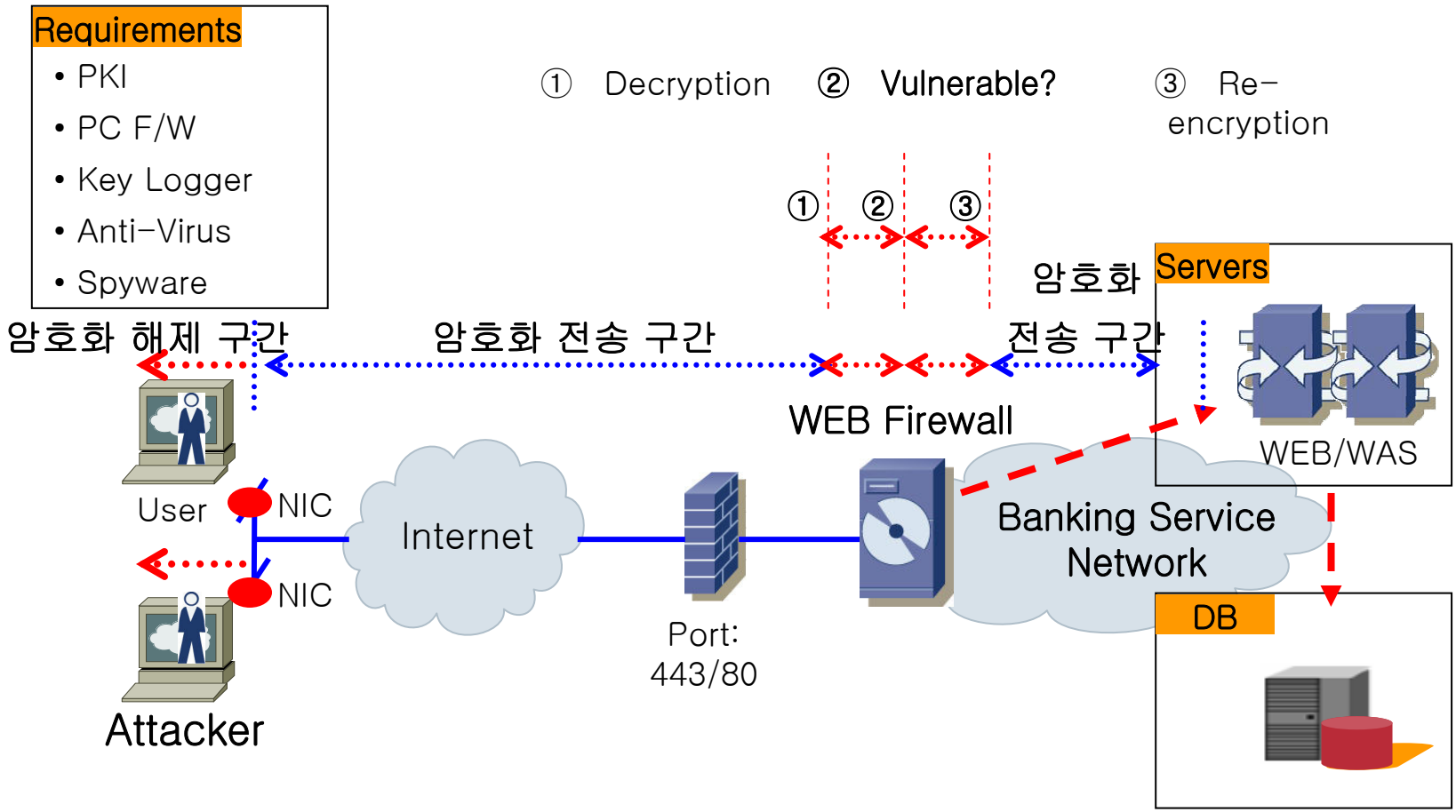

**75%** 이상의 사이버공격과 인터넷 보안침해 사고는 인터넷 웹 애플리케이션 취약점을 이용하여 발생한다.

**-Gartner Group Report 2002-**

# Web Firewall #1



대규모 네트워크를 운영하는 기업 및 금융권에서는 많은 수가 PKI, SSO/EAM을 사용하고 있어, 실시간 암호 해제, 판단, 재 암호화를 해야 취약점 방어를 할 수 있는 웹 방화벽 아키텍처는 비효율적일 수 있음.



# Web Firewall #2



**Threat**

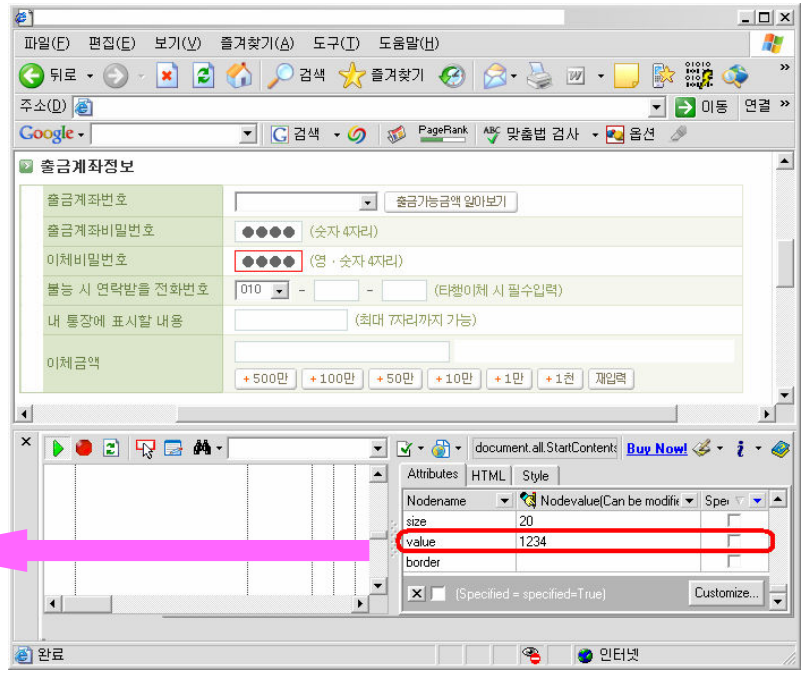
- IE DOM 조작 기법

- 이러한 방식의 취약점이 인터넷 접속 다양성을 저하시키는 가장 큰 보안
- 중요정보 열람/조작 가능
- Script kids 용 자동화 도구 배포시작 됨

## 사례1: IE DOM 메모리 조작

- 1998년경 DOM 개발 툴 발표
- 2000년경 고급 해커들 사이에서 이슈화
- 2004년경 애드웨어에서 대중적으로 사용
- 2007년 웜/트로이목마 형태 제작 가능성

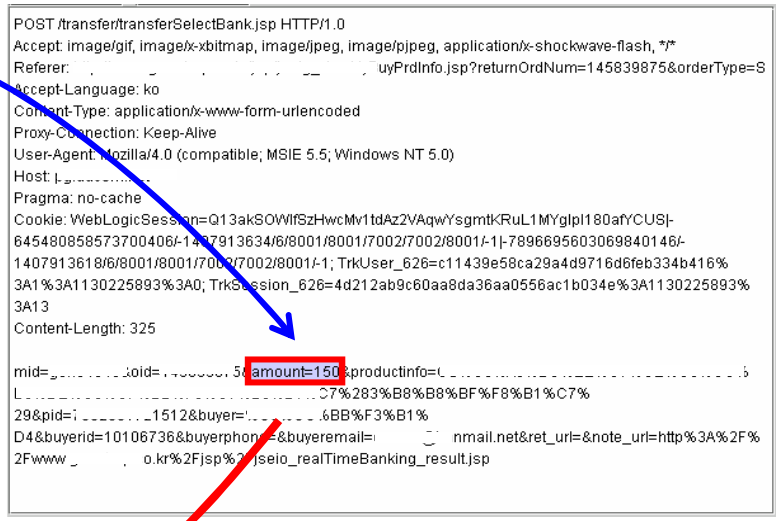
기존 키로깅 방식이 아닌 **DOM** 메모리 상에서 사용자가 입력한 계좌비밀번호 '**1234**'를 직접 열람



# Web Firewall #3



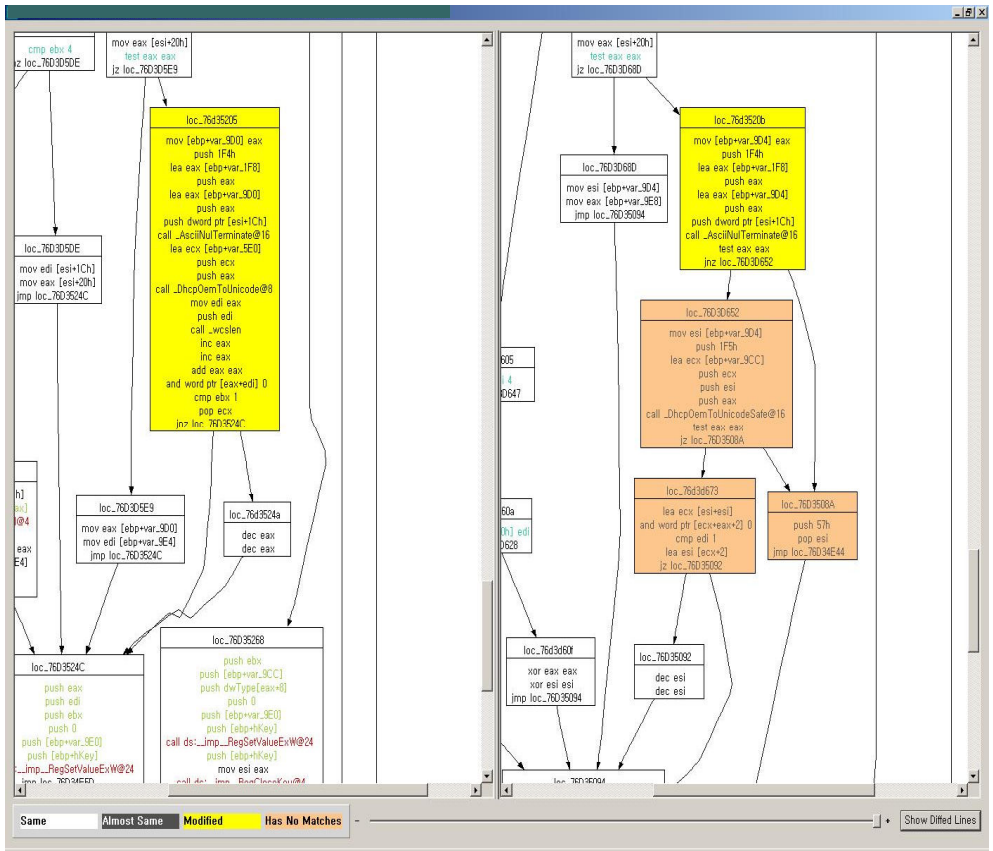
특정 인자 값을 Proxy를 통해 위·변조하는 경우, 네트워크에서 감시하는 웹 방화벽의 경우 이를 전혀 감지할 수 없음. 따라서 인터넷을 통한 결재 서비스는 클라이언트에서의 보안성이 유지되지 못할 경우, 대단히 위험함.

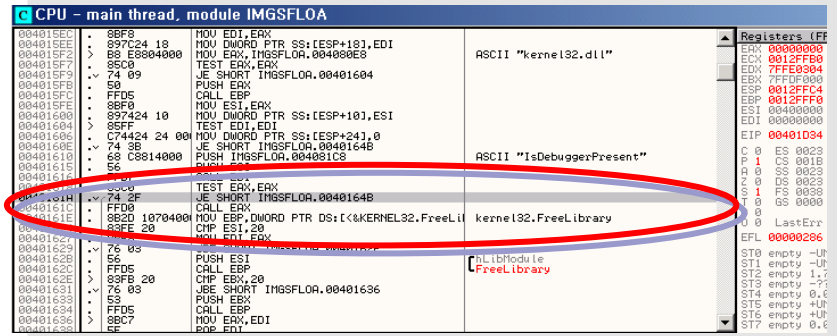



# Web Firewall #4

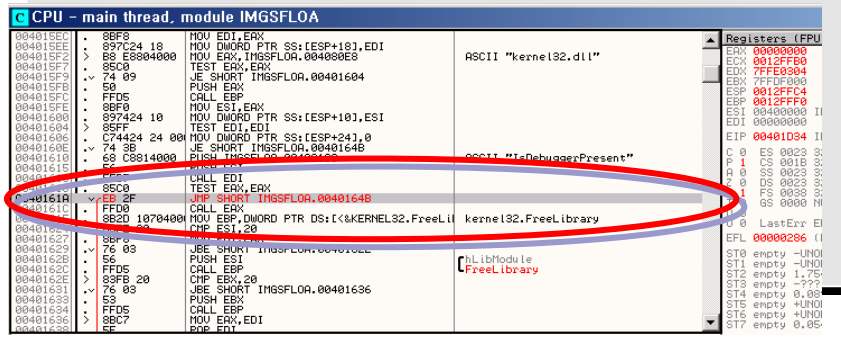


Server Side에서 배포한 ActiveX 취약점을 이용하는 경우와 Reverse Engineering을 통해 정교하게 위조된 Client 보안체계 우회를 통해 서버 측 취약점 공격이 가능. 이때 웹 방화벽은 세션과 무관한 공격이므로, 공격을 탐지할 수 없음.(웹 방화벽에서 방어 불가능)




- JE → JMP / 인증 중요 로직 우회



# Web Firewall #5



동일한 표현을 URL, Unicode, %uXXYY, Chunked, Multipart/form-data, application/x-www-form-urlencoded, Multiple Encoding 등 대단히 다양하게 표현할 수 있으며, 이를 완전하게 막는 것은 결국 웹 방화벽 기술이 아닌 기술 엔지니어의 방어 기술 구현 능력에 따라 편차가 존재.

• **Normal XSS**

```
<SCRIPT>alert("XSS")</SCRIPT>
<IMG SRC="javascript:alert('XSS');">
<IMG SRC=javascript:alert('XSS')>
```

• **Evasion XSS**

```
<IMG SRC=JaVaScRiPt:alert('XSS')>
<IMG SRC=javascript:alert(&quot;XSS&quot;)>
<IMG SRC="jav&#x0D;ascript:alert('XSS');">
<IMG SRC=" &#14; javascript:alert('XSS');">
<IMG
SRC=&#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#10
8;&#101;&#114;&#116;&#40;&#39;&#88;&#83;&#83;&#39;&#41;>
<IMG
SRC=&#0000106&#0000097&#0000118&#0000097&#0000115
```

등 수 없이 많은 조합이 가능

# Web Firewall #6



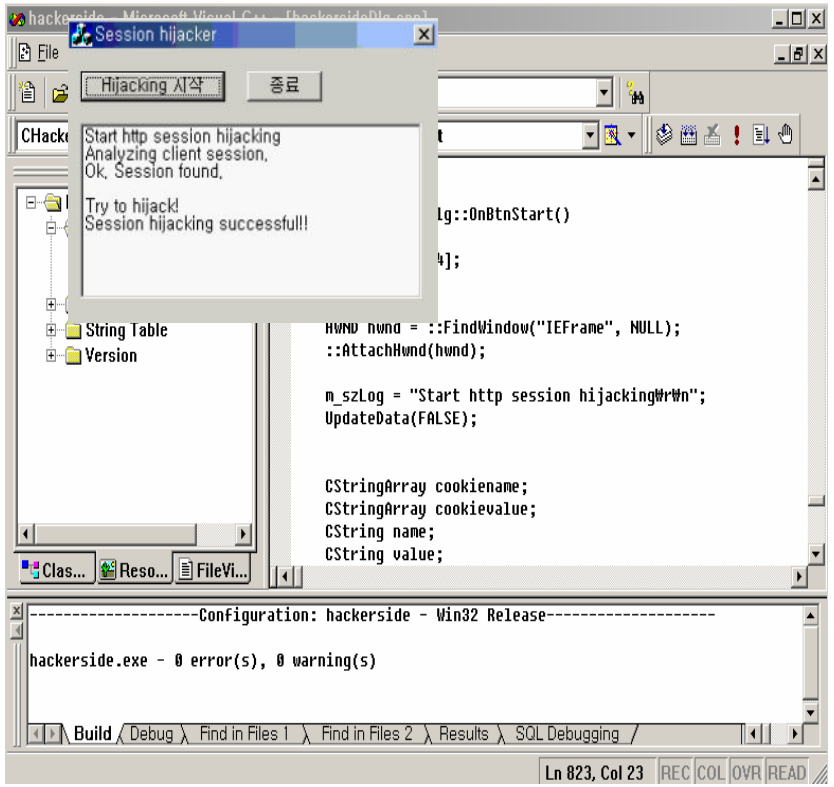
웹방화벽이 모든 세션을 감시하는 구조가 아닌 이상, Session Hijacking 기법을 이용, 타 사용자의 세션을 이용할 경우, 통과됨. 모든 세션을 감시하게 될 경우, 웹 방화벽 자체의 부하량 증가로 인해 전체 서비스의 효율성이 저하되는 역기능이 예상. 결국 웹 방화벽에서 방어 불가능

**Client 1**

```
Cookie: sess=TWGYLZIAAACVDQ3UUSZQV2I
```

**Client 2**

```
Cookie: sess=TWGY0WYAAACVFQ3UUSZQV2I
```



# Anti-Reverse Engineering



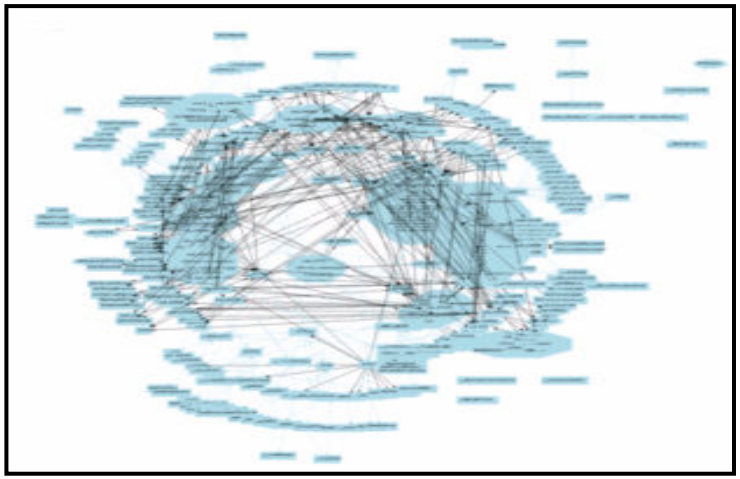
**Project**

- Code obfuscation
- Packing
- Runtime encryption/decryption
- Random garbage code insertion
- Virtualization
- Isolation

디스어셈블

```
UPX1:0044A4D0
UPX1:0044A4D0                cmp     [esp-0Ch+arg_10], 1
UPX1:0044A4D5                jnz     loc_4AA66C
UPX1:0044A4DB                pusha
UPX1:0044A4DC                mov     esi, offset dword_431000
UPX1:0044A4E1                lea     edi, [esi-30000h]
UPX1:0044A4E7                push    edi
UPX1:0044A4E8                or      ebp, 0FFFFFFFFh
UPX1:0044A4EB                jmp     short loc_44A4FA
UPX1:0044A4ED ; ÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇ
UPX1:0044A4ED                align 10h
UPX1:0044A4F0
UPX1:0044A4F0 loc_44A4F0:                            ; CODE XREF: start:loc_44A501↓j
UPX1:0044A4F0                mov     al, [esi]
UPX1:0044A4F2                inc     esi
UPX1:0044A4F3                mov     [edi], al
UPX1:0044A4F5                inc     edi
UPX1:0044A4F6
UPX1:0044A4F6 loc_44A4F6:                            ; CODE XREF: start+D2↓j
UPX1:0044A4F6                                        ; start+E9↓j
UPX1:0044A4F6                add     ebx, ebx
UPX1:0044A4F8                jnz     short loc_44A501
UPX1:0044A4FA
UPX1:0044A4FA loc_44A4FA:                            ; CODE XREF: start+1B↑j
UPX1:0044A4FA                mov     ebx, [esi]
UPX1:0044A4FC                sub     esi, 0FFFFFFFCh
UPX1:0044A4FF                adc     ebx, ebx
UPX1:0044A501
UPX1:0044A501 loc_44A501:                            ; CODE XREF: start+28↑j
UPX1:0044A501                jb      short loc_44A4F0
UPX1:0044A503                mov     eax, 1
UPX1:0044A508
UPX1:0044A508 loc_44A508:                            ; CODE XREF: start+62↓j
UPX1:0044A508                add     ebx, ebx
UPX1:0044A50A                jnz     short loc_44A513
UPX1:0044A50C                mov     ebx, [esi]
UPX1:0044A50E                sub     esi, 0FFFFFFFCh
UPX1:0044A511                adc     ebx, ebx
UPX1:0044A513
UPX1:0044A513 loc_44A513:                            ; CODE XREF: start+3A↑j
UPX1:0044A513                adc     eax, eax
```

함수호출관계 분석



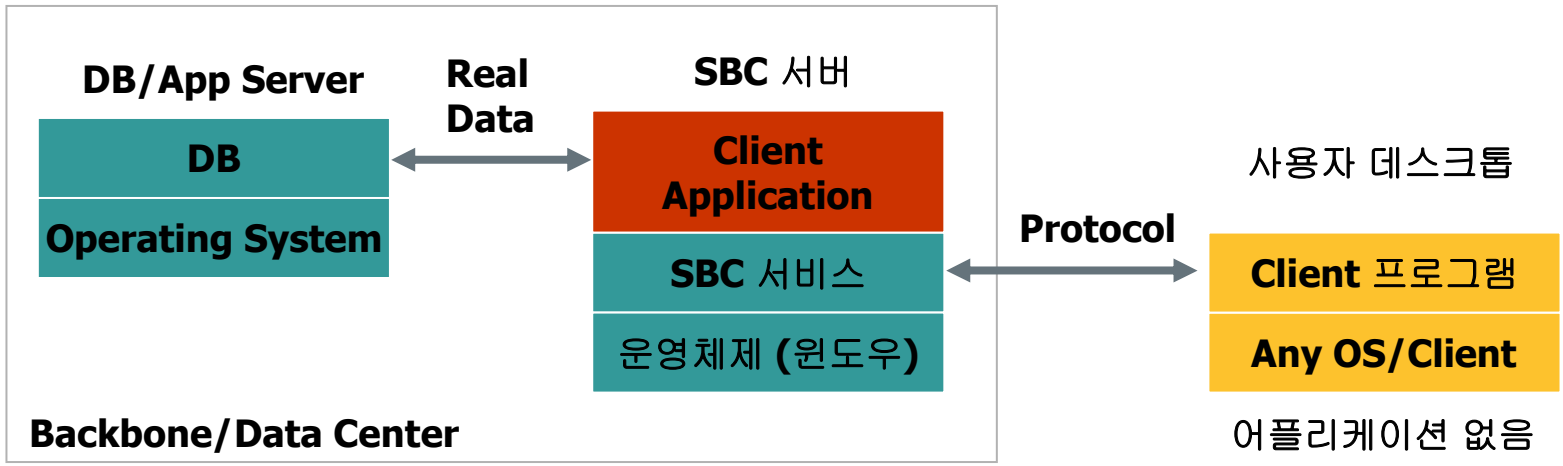
# SBC(Server Based Computing) #1



업무 수행을 위한 인트라넷과 인터넷 동시 이용은 불가피하며, 이에 따른 바이러스 및 웜의 유입과 강력한 보안이 요구되는 내부 기밀자료의 유출을 막기 위해 Thin Client는(SBC, Server Based Computing) 업무용도 전용 단말기 수준의 보안 개념을 갖고 있음.

Thin Client 사용에 따른 TCO 분석(Source : 美 Intelliquest社)

- 네트워크 관리 비용 절감(55%)
- 동료의 기술 문제를 돕기 위해 소요하는 시간 등 "비공식적인 관리"의 감소(14%)
- 하드웨어 구입 비용 절감(13%)
- 용이한 응용 프로그램 소프트웨어 업그레이드 배포(11%)
- 운영 체제 업그레이드 배포 비용 절감(3%)

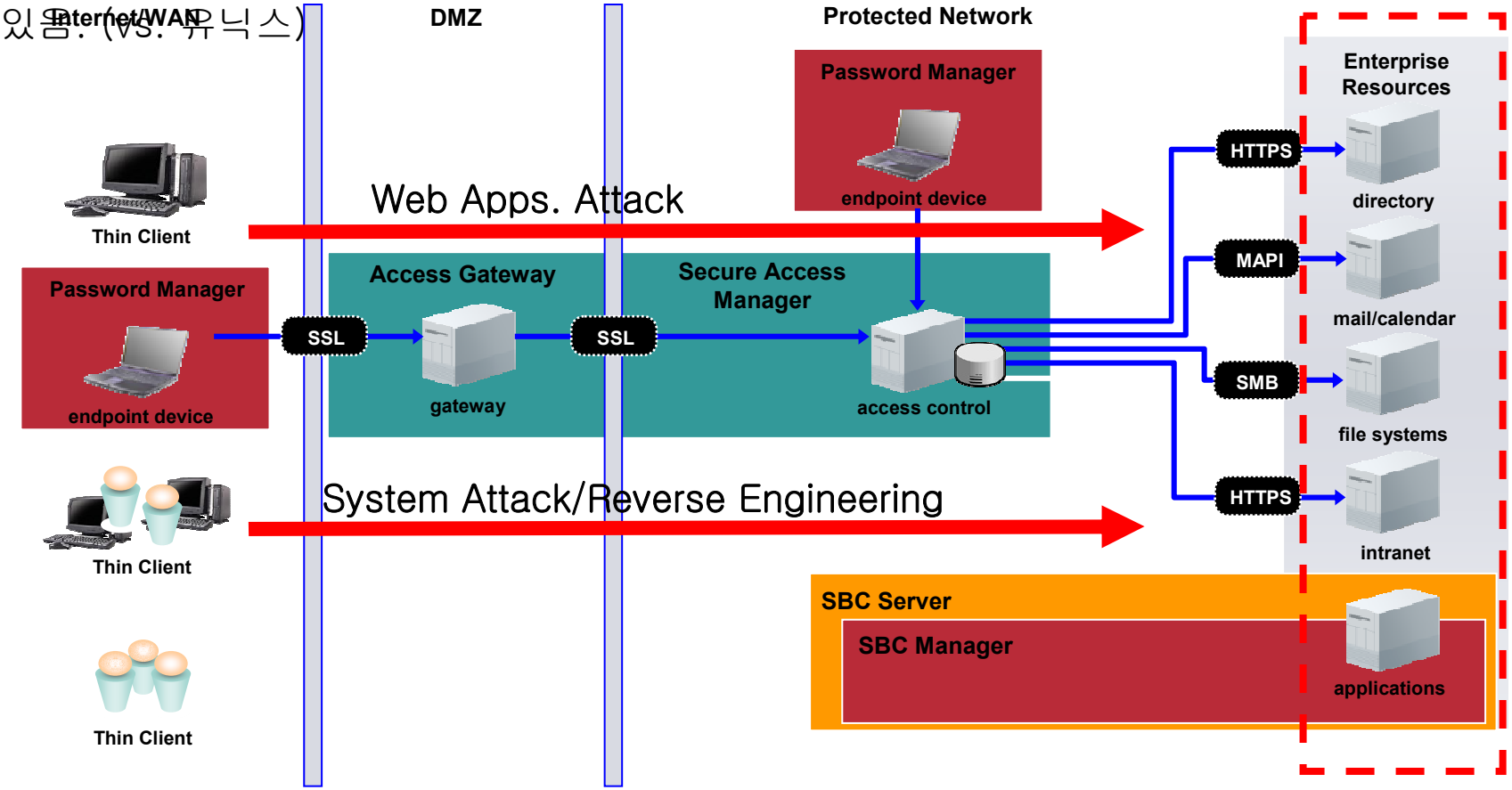



Thin Client (Server Based Computing)의 개념도

# SBC(Server Based Computing) #2



Thin Client의 기술은 관리비용 절감에 효율적이나, Thin Client 기술이 기존의 중요 자원이 존재하는 각종 파일서버, 그룹웨어의 중요 데이터에 대한 악의적 접근 및 공격에서 자유로울 수 있는 기술은 결코 아니며 오히려 해커에게 종합선물세트를 제공하는 결과를 초래할 수도 있음. (vs. 유닉스)

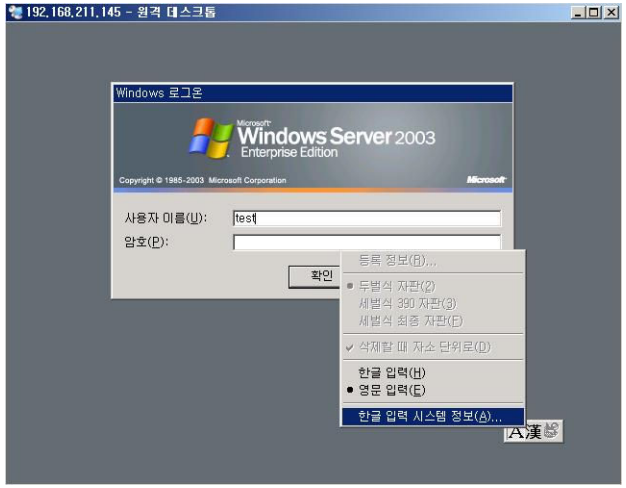


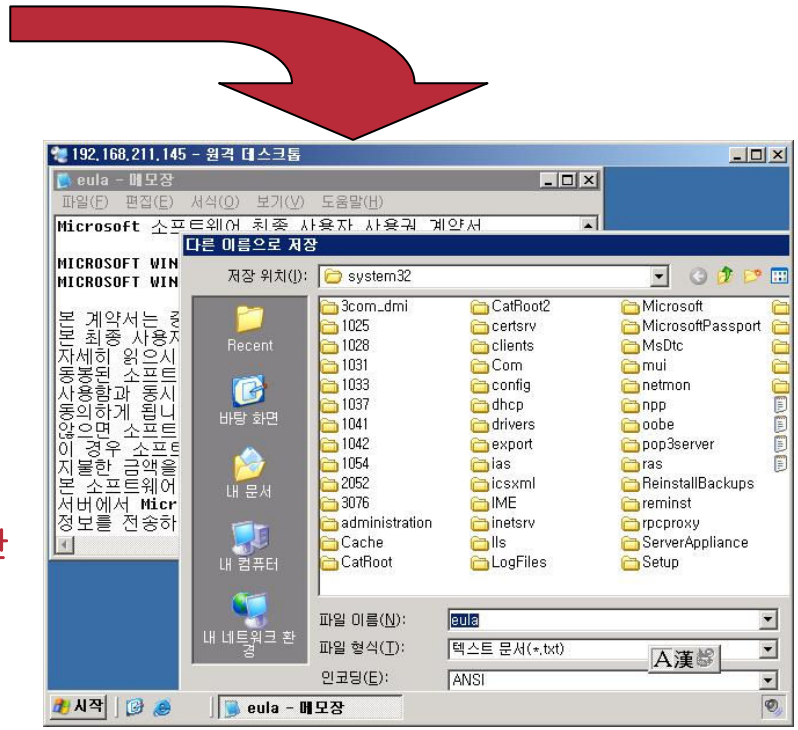
# SBC(Server Based Computing) #3



SBC는 local privilege elevation 공격에 취약하다. 서버에 설치된 OS, 백신프로그램, 서비스 어플리케이션 상당수는 privilege elevation 취약점이 존재하며 이들 취약점은 일반 컴퓨팅 환경이 아닌 SBC 환경에선 파급도가 매우 큼. (일반사용자권한 -> 관리자권한 획득 가능)

**이 경우 권한이 없는 모든 문서에 대한 유출이 가능하므로 보안 위협이 오히려 증가할 수 있음**




일반사용자가 마우스 클릭만으로 관리자권한 획득 가능한 취약점 발견

# Learning Effect



| 공격 유형 | 방어대책 존재여부 | 현행 방어 대책 | 비고 |
|---|---|---|---|
| Process/COM/DCOM/COM+ Debugging/Reverse Engineering | X | 어플리케이션 보호를 위한 대책은 Secure Programming 이외의 대책이 존재하지 않음 | |
| 암호화 해독 방지(DOM Inspection 방지) | X | DOM Inspection을 통한 암호화 해독 방지 기능에 대한 대책이 존재하지 않음 | |
| API Hooking 공격 방지(DLL Injection 계열) | X | 각종 Injection 계열의 공격 방어를 위한 대책이 존재하지 않음 | |
| SQL Injection 공격 방어 | △ | 웹 방화벽을 통한 공격 방어가 가능하나 다양한 유회 공격 기법에 완전하게 대처하는 것은 불가능 | 웹 방화벽 도입으로 인한 네트워크 속도저하 부작용 |
| XSS 공격 방어 | △ | | |
| ActiveX 컨트롤 취약점 공격 | X | ActiveX 컨트롤 취약점이 발견될 경우, 해당 조직내의 모든 PC에 대한 권한 획득이 가능하나, 이의 대책은 전무함 | |
| COM 후킹방식 키로거 방어 | X | 어플리케이션 특화 키로거 방어 대책은 존재하지 않음 | |
| Proxy을 통한 인자값 위변조 공격 | X | 웹 방화벽으로도 Proxy를 통한 인자값 위변조 공격의 방어는 불가능함 | |
| Session hijacking 공격 | X | 웹 방화벽으로도 세션 Hijacking 공격에 대한 방어는 불가능 | |
| Regmon, filemon에 대한 방어 대책 | X | 어플리케이션 악의적 분석을 위한 방어대책 전무함 | |

# III. Understanding Virtualization



- Why use virtual machine?
- Virtualization Approaches
- Licensing & Distribution
- Well-Known Virtualization Architecture
- VMAppSaver I/O Control
- VMAppSaver 실행 화면

# Why Use Virtual Machines?



- Support heterogeneous applications using one physical machine (both Windows and Linux apps)
- TCO(Total Cost of Ownership) Savings
- Allows the execution environment to be tailored to the application
- Cost down for web hosting service
- Easy to manage(S/W Upgrade, Update etc…)
- Secure sandboxes for untrusted applications
- By managing QoS
- Make system mobility easier to implement.

# Virtualization Approaches



- **Single OS image: Virtuozo, Vservers, Zones**
  - **Group user processes into resource containers**
  - **Hard to get strong isolation**
- **Full virtualization: VMware, VirtualPC, QEMU**
  - **Run multiple unmodified guest OS's**
  - **Hard to efficiently virtualize x86**
- **Para-virtualization: UML, Xen**
  - **Run multiple guest OS's ported to special arch**
  - **Arch Xen/x86 is very close to normal x86**

# Licensing & Distribution



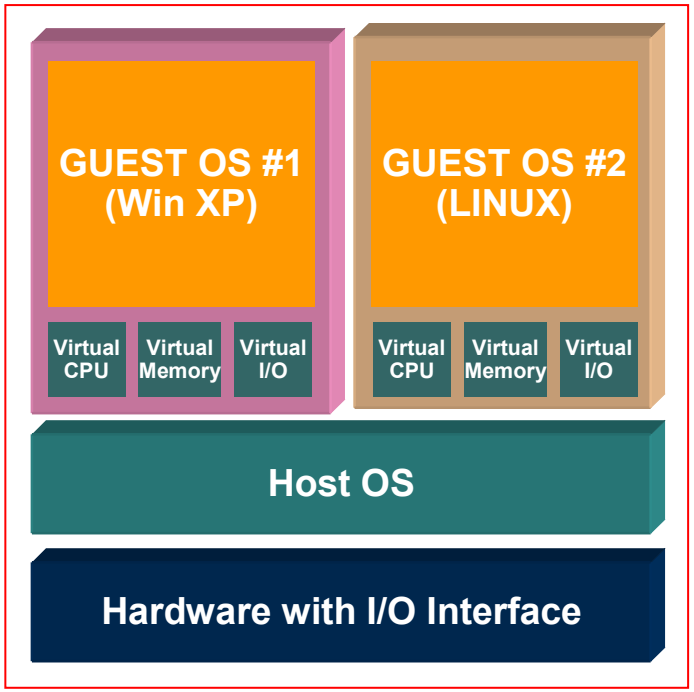
- **License**
  - **Open source (Xen, UML)**
    - **Visible effects of open source community at work**
  - **Commercial (VMware)**
    - **Also, XenSource**
- **Distribution**
  - **Para-virtualization requires kernel modifications**
    - **Xen is (or soon to be) a part of multiple distributions: Fedora Core 4, Debian, unofficial: Gentoo, Mandrake and SUSE distributions**
  - **Privilege**
    - **Xen (root, patch kernel, domain 0 privileges setup)**
    - **VMware Workstation (root, installation only)**
    - **UML: user-level**



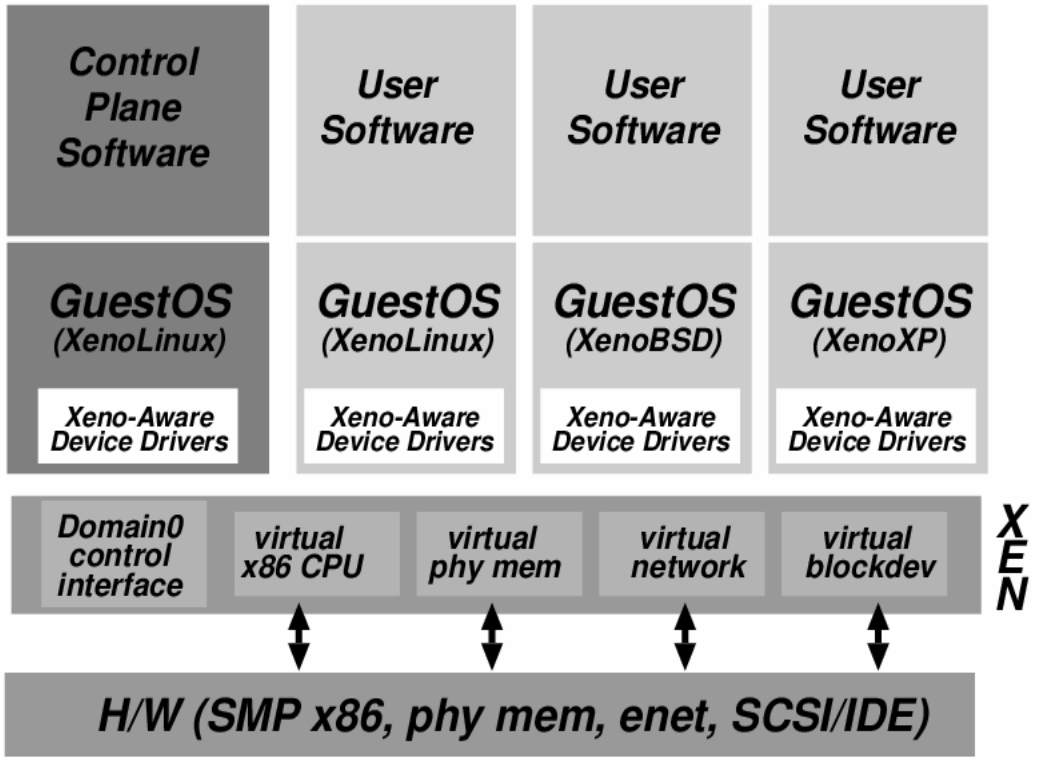
**Source :** Kate Keahey, Argonne National Laboratory

# Well-Known Virtualization Architecture






**VMware Architecture**




**XEN Architecture**

Source : Xen and the Art of Virtualization (Xen 1.x.)

Published at SOSP 2003

# Problems



- **Virtualization == Higher Security???**
  - 가상화 기술과 보안성 향상은 별개의 문제
  - **Registry, File System, Object** 등에 대한 방어 필요

- **COM/DCOM Virtualization**
  - 일부 어플리케이션의 완전한 실행을 위해서 구현 필요
  - 다양한 어플리케이션 환경에 적응하는 시간이 필요

- **OS** 종속적
  - **Client** 가상화 기술을 위해서는 **OS** 종속적 구현은 불가피
  - 서버 가상화를 통한 **Streaming** 서비스와 같은 대안은 존재

# IV. Secure Virtualization Architecture



- Why use virtual machine?
- Virtualization Approaches
- Licensing & Distribution
- Well-Known Virtualization Architecture
- VMAppSaver I/O Control
- VMAppSaver 실행 화면

# Three Architecture



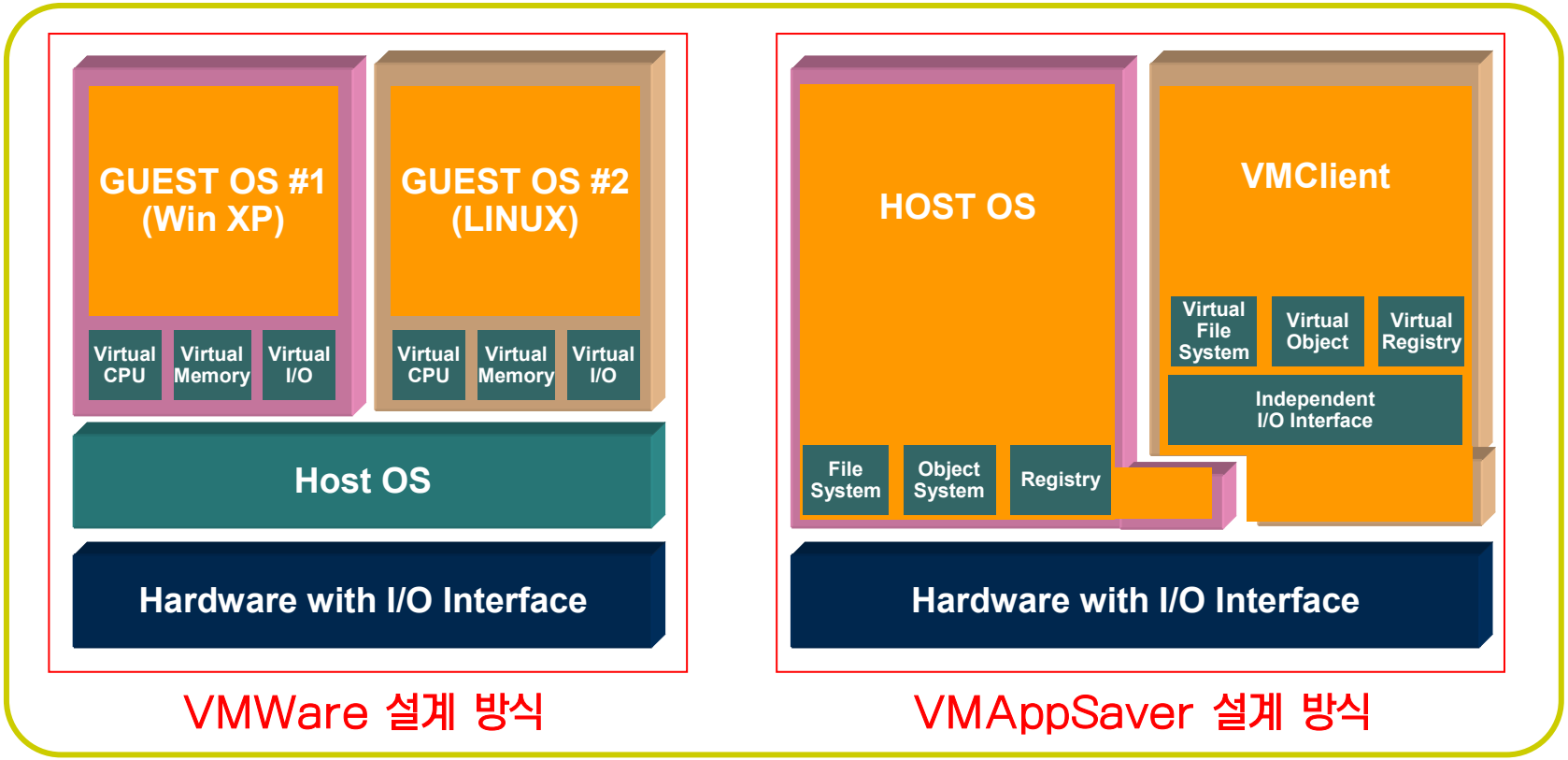
- **VMAppSaver Architecture**
  - 클라이언트 어플리케이션의 보안 극대화가 목표
  - 악의적 어플리케이션 분석 방지

- **VM VAC Architecture**
  - 업무공간과 인터넷 공간의 분리
  - 인터넷을 통한 정보 유출의 방지

- **VMWebSaver Architecture**
  - 웹 방화벽이 막을 수 없는 웹 취약점의 방어
  - 서버 가상화를 통한 **Streaming** 서비스와 같은 대안은 존재

# VMAppSaver Architecture

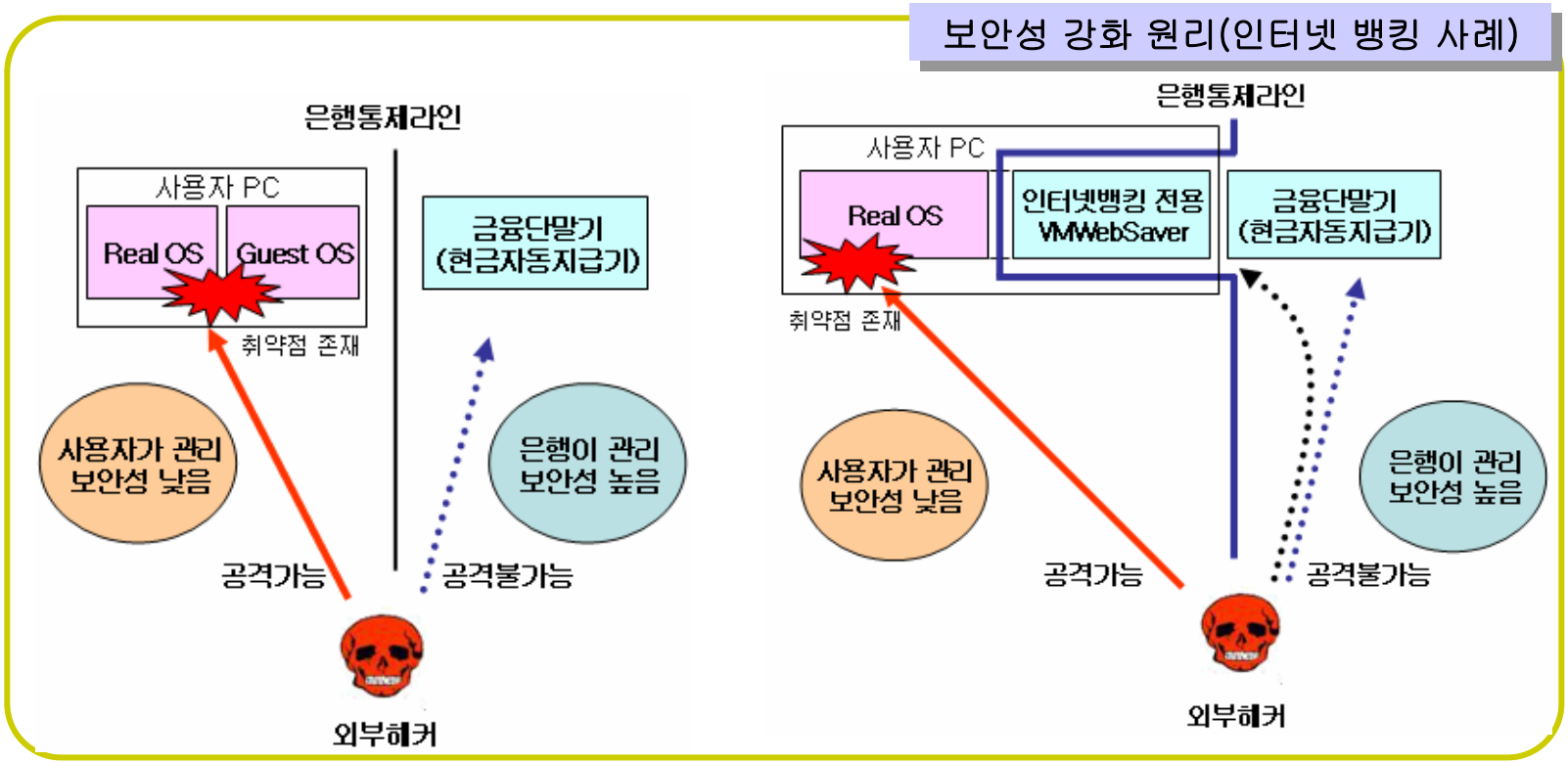




# VMAppSaver 통제의 개념



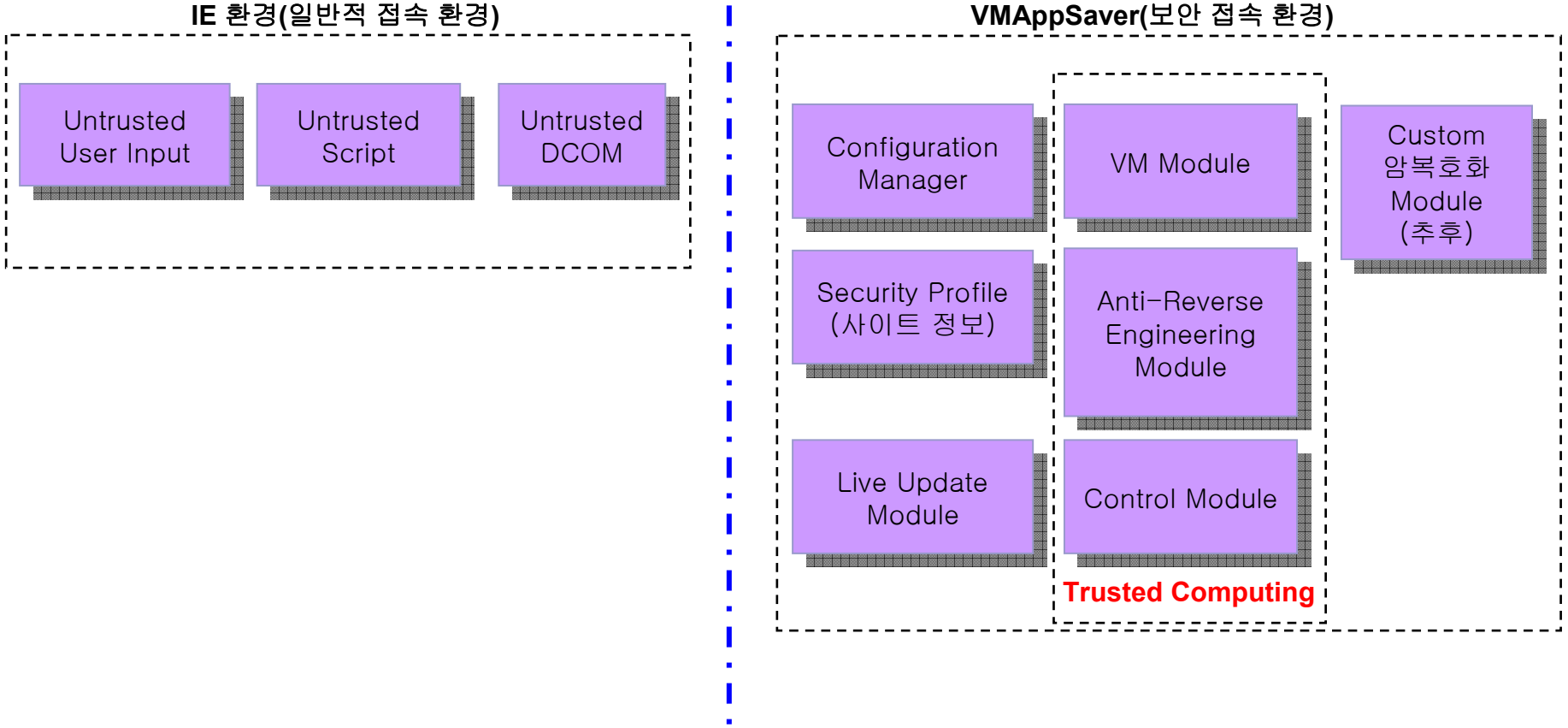
서비스 제공자가 사용자의 PC 환경이나 보안수준에 대한 고려 없이 보안수준을 획기적으로 높일 수 있도록 사용자가 서비스를 사용하는 순간에 PC의 가상화 공간을 이용하기 때문에, 보안수준을 획기적으로 높일 수 있음.



# VMAppSaver Module Architecture

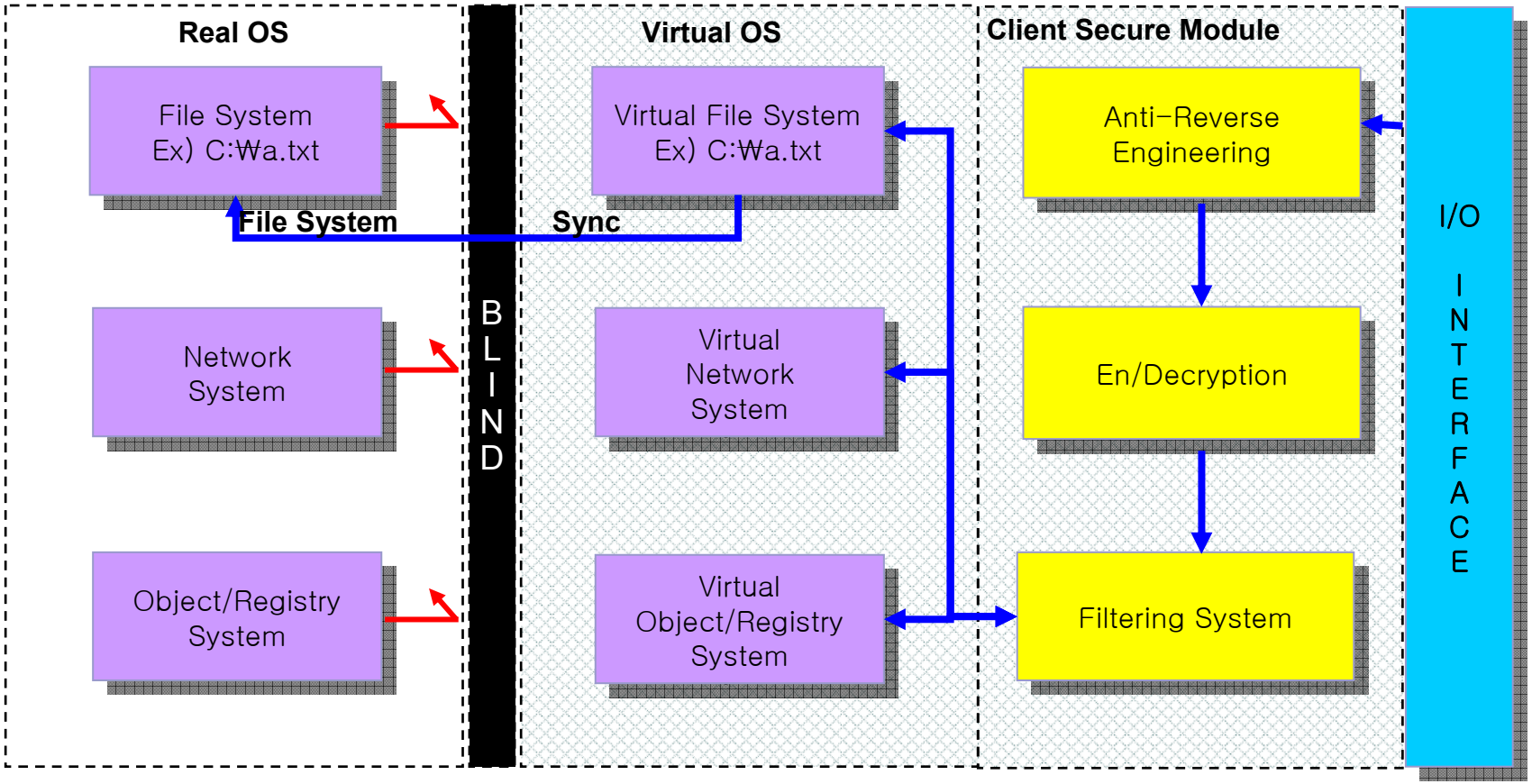


일반적인 IE 환경에서는 신뢰되지 않는 User Input, Script, DCOM등에 대한 접근이 가능하기 때문에 결과적으로 이를 악용할 경우, 침해사고를 유발하는 요인이 됨. VMAppSaver 환경에서는 서비스 제공자가 배포한 Apps.에 대해 Anti-Reverse Engineering, Anti-Debugging 등의 조작이 불가능하도록 가상화 기술을 통한 제어를 구현하고 있음.



# VMAppSaver I/O Control

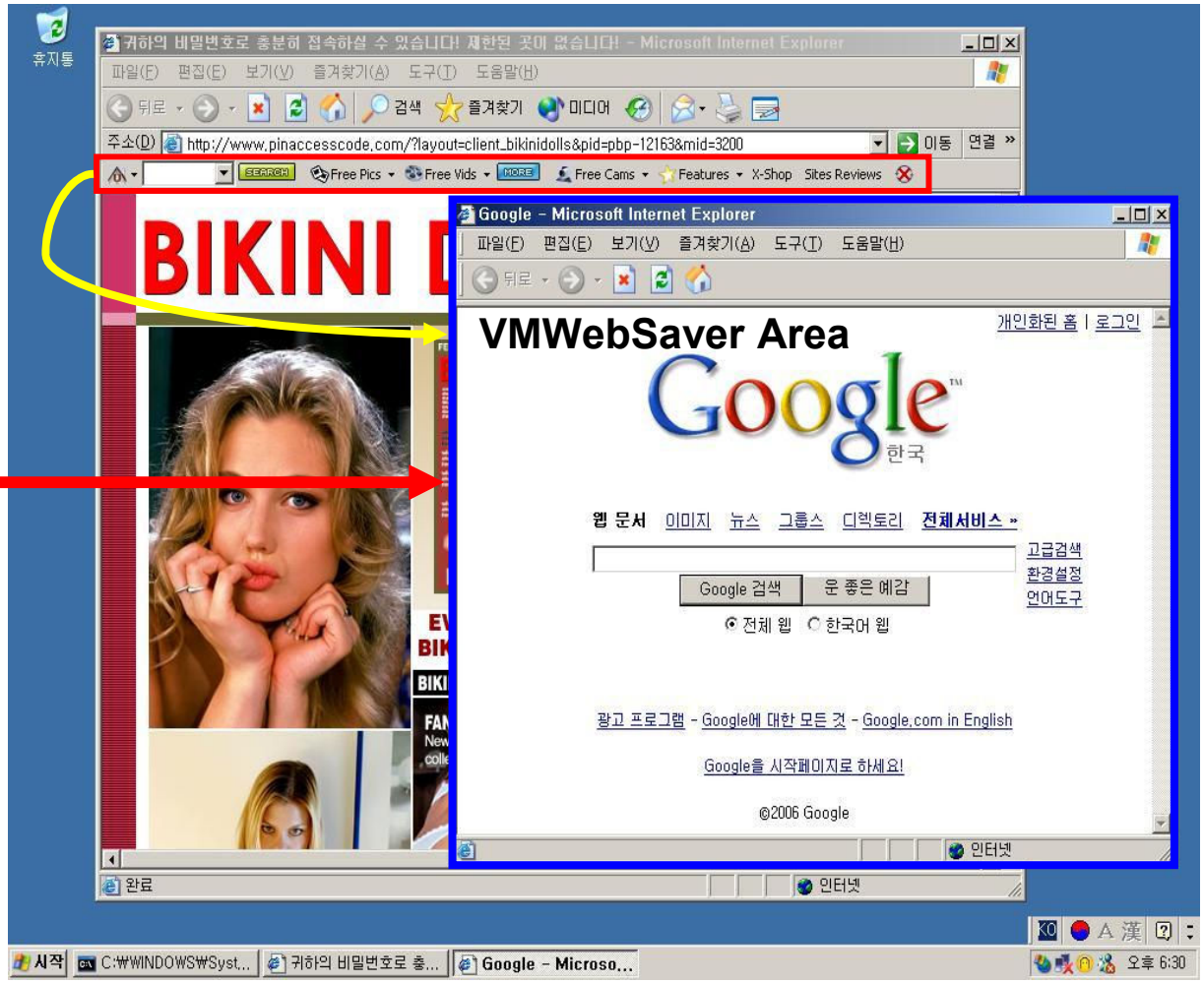




# VMAppSaver 보호 기능 #1



IE와 같은 웹 서비스를 위한 브라우저의 경우, 변조되지 않은 순수한 형태의 IE를 띄워, 실제 공간에 악성 바이러스 또는 애드웨어 등에 감염이 되어 있어도 가상화 공간의 IE에는 영향이 전혀 없어, 뱅킹 서비스를 게임방 등의 공공장소에서도 얼마든지 이용할 수 있음.




이미 악성사이트 방문으로 인해 애드웨어에 감염된 Tool Bar 형태 악성 코드의 영향을 VMWebSaver가 방어하고 있음(초기 화면도 원래대로 환원)

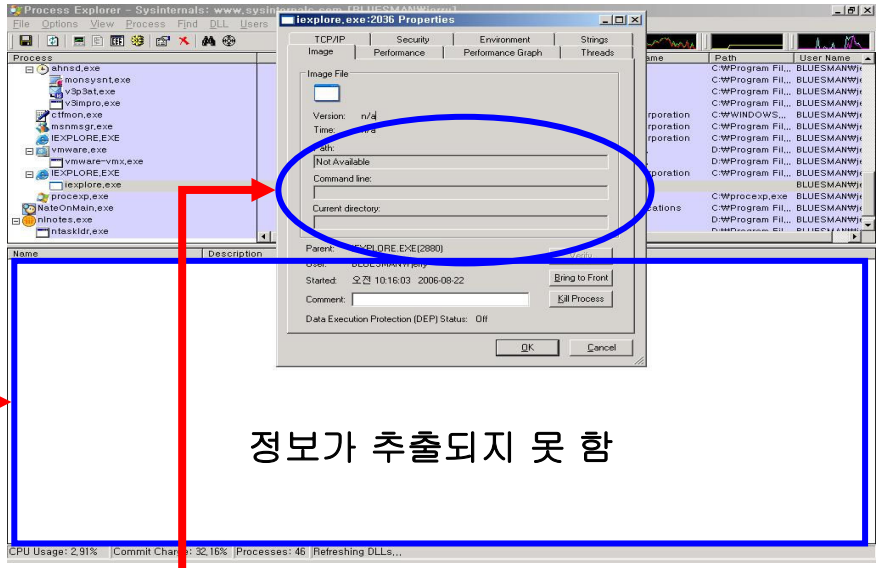
# VMAppSaver 보호 기능 #2



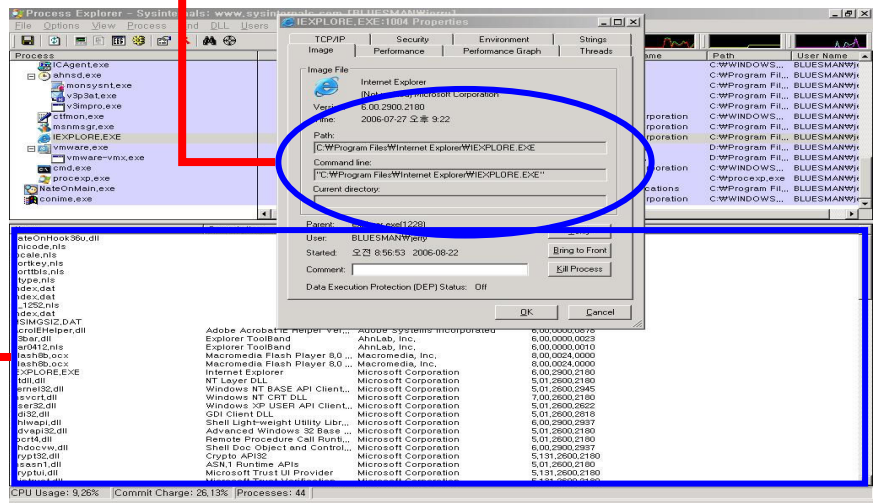
VMWebSaver는 가상화된 독립적 프로세스에서 동작되기 때문에 프로세스와 핸들과 같은 목록화가 불가능.

악의적인 공격자가 특정 Application에 대한 분석을 통해 전자적 침해행위를 원천적으로 방어할 수 있도록 설계되어 있음.

인터넷 금융서비스 제공을 위한 IE 또는 전용 Apps.에 적용될 경우, 보안성이 대폭적으로 향상되는 효과가 있음.




VMWebSaver 에 의한 독립적 프로세스(After)



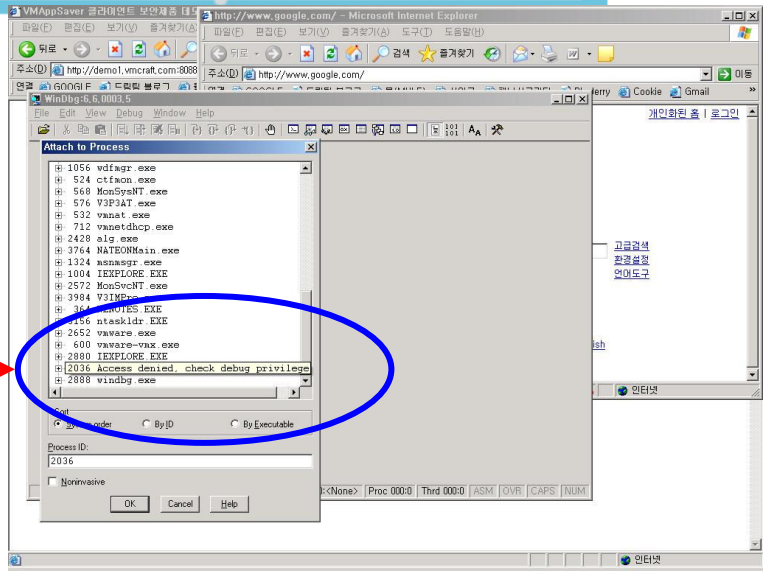
일반적 환경에서의 OS 종속적 프로세스(Before)

# VMAppSaver 보호 기능 #3

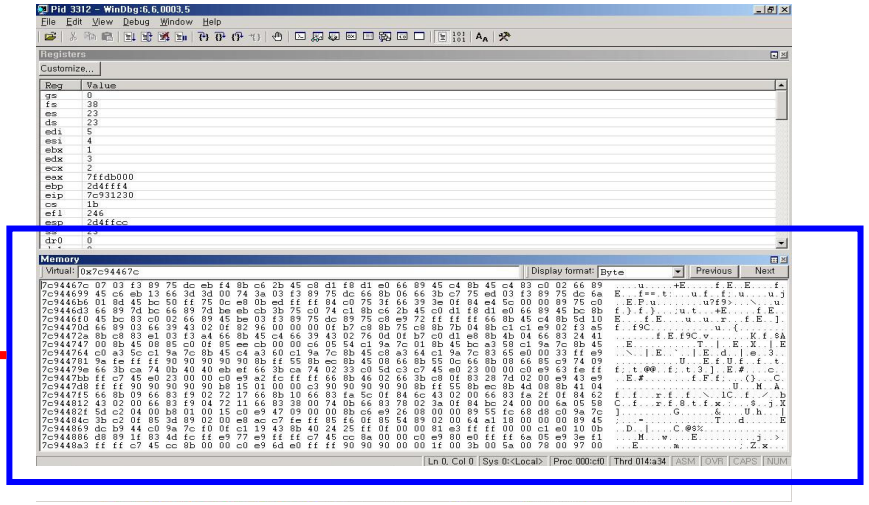


VMWebSaver는 디버거 등을 통한 분석 자체를 불가능하게 하여, API Hooking(DLL Injection 포함), ActiveX Control 취약점 등에 효과적인 대응이 가능 함.

공격자는 특정 Application에 대한 분석이 선행되고, 이를 악용하여 악의적이고 지능적인 공격이 가능할 수 있으나, VMWebSaver 환경에서는 이러한 일련의 분석 과정 자체가 불가능 함.




**VMWebSaver** 에 의해 디버거 삽입 방지(After)



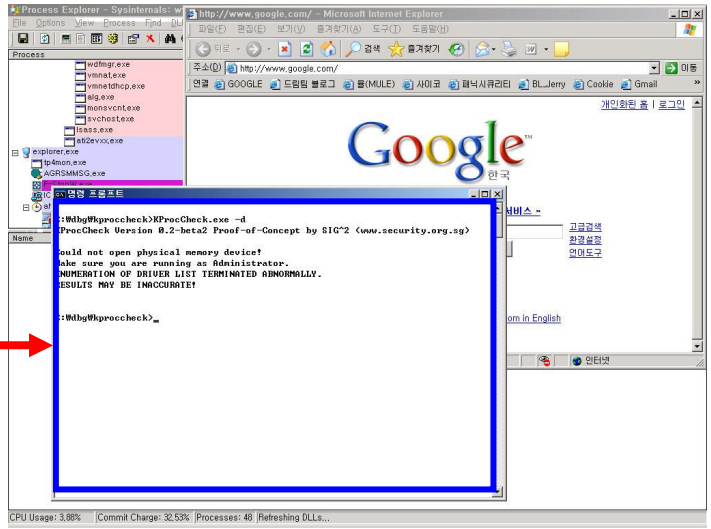
일반적 환경에서의 어플리케이션 디버거 분석 가능(Before)

# VMAppSaver 보호 기능 #4

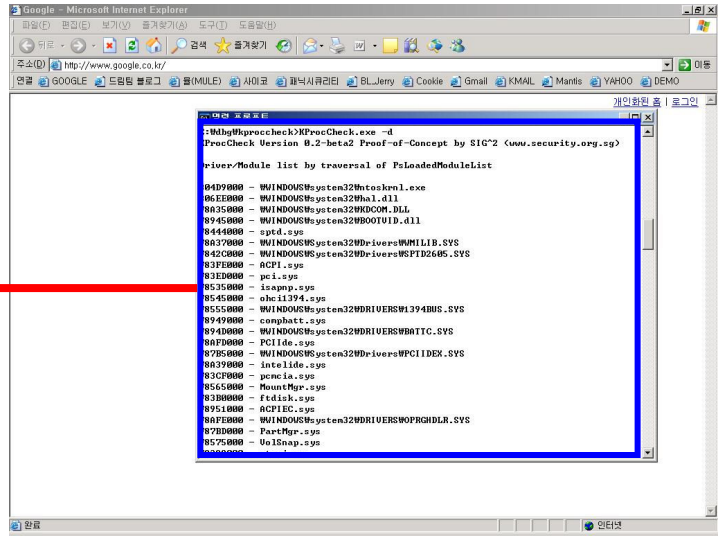


VMWebSaver는 공격자의 지능적 공격을 방어하기 위하여 Physical Memory에 대한 방어 기능이 부가적으로 존재하며, 이를 통해 공격자의 Application 분석 자체를 못하도록 방어 함.

일반적 OS 위에서 구동되는 Application의 경우, Physical Memory에 대한 방어가 불가능하여, 이를 통해 공격자는 순간 저장되는 Transaction Data 등을 가로챌 수 있음.



**VMWebSaver** 에 의한 Physical Memory 보호(After)

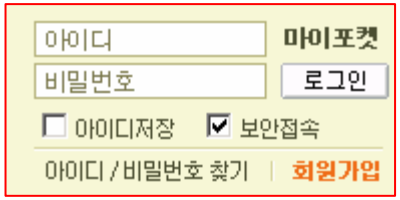
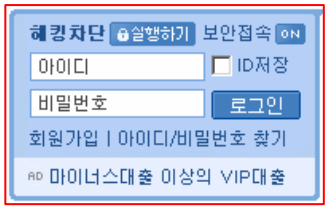
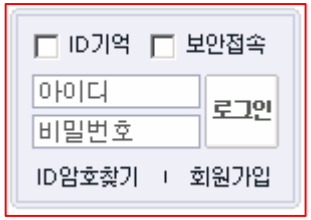


일반적 환경에서의 Physical Memory 목록화 가능(Before)

# 국내 포탈 서비스의 보안 강화 현황

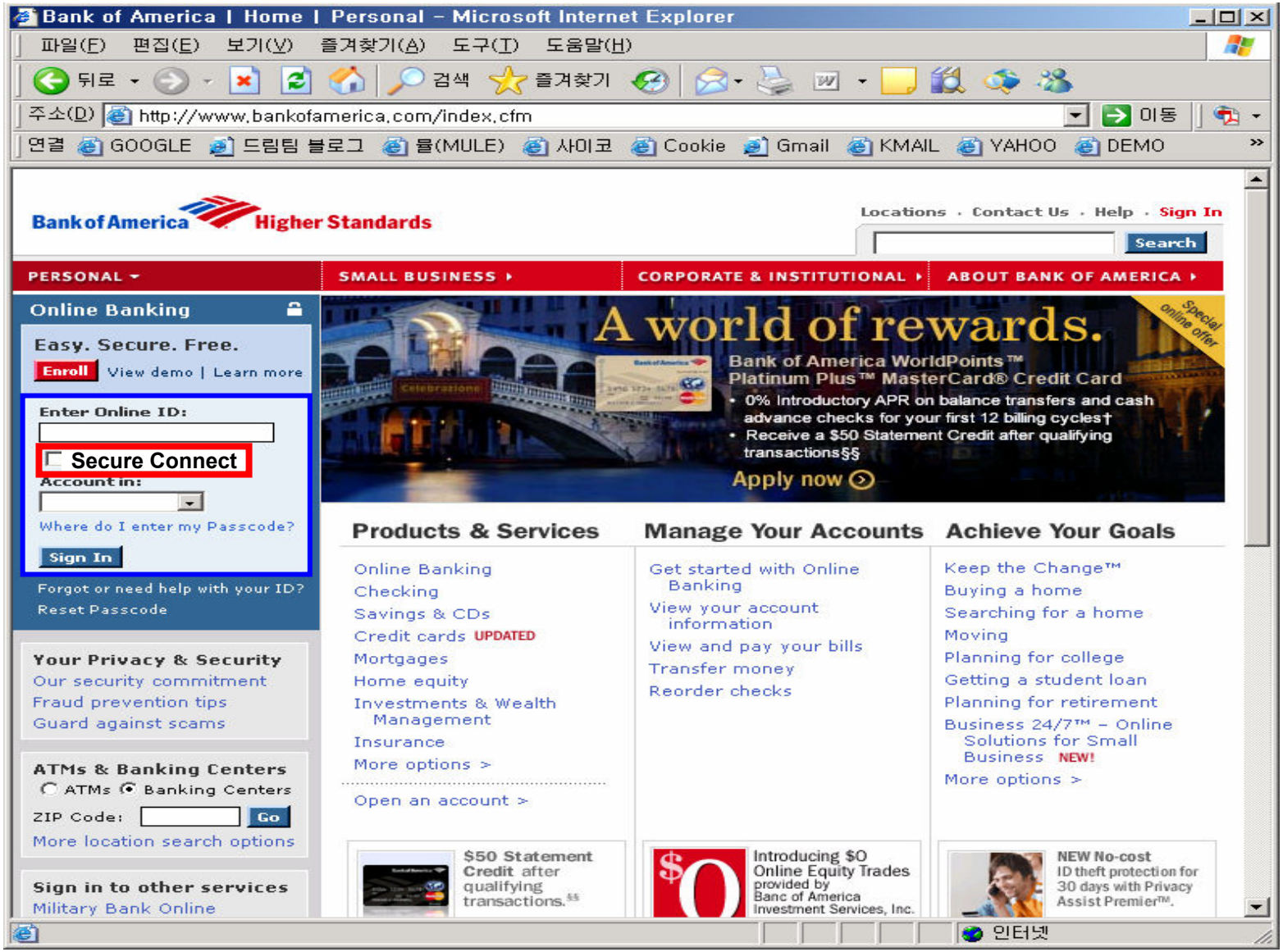


국내 포탈 서비스의 경우 대부분이 “보안접속” 서비스 제공을 통해 보다 차별화된 인증 서비스를 제공하기 위한 노력을 계속하고 있음. 그러나 “보안접속” 서비스 구성요소는 기본적인 보안 서비스만을 제공할 뿐이며, 실제적인 “보안접속”효과는 미미한 실정임.

| 인터넷 포탈 구분 | 구현 방식 | 현재 구현 |
|---|---|---|
| NAVER | 기존에는 NAVER의 로그인 서비스가 보안 접속과 관련이 없는 일반 Plain Text 전송 방식이었으나, 시범 서비스 기간 이후 보안 접속 서비스는 기본적으로 제공 되는 서비스임 |  |
| DAUM | DAUM의 서비스 역시 NAVER와 마찬가지로 시범 서비스를 일정 기간 정도 서비스 한 후 현재는 기본적으로 제공되는 서비스 임 |  |
| empas | EMPAS는 기본적으로 보안접속 서비스가 제공되지는 않지만, 수동으로 설정하면 보안접속 서비스를 제공함. NAVER, DAUM의 사례를 볼 때, 추후 기본 접속 서비스로 제공될 것 임. |  |

# VMAppSaver 예제



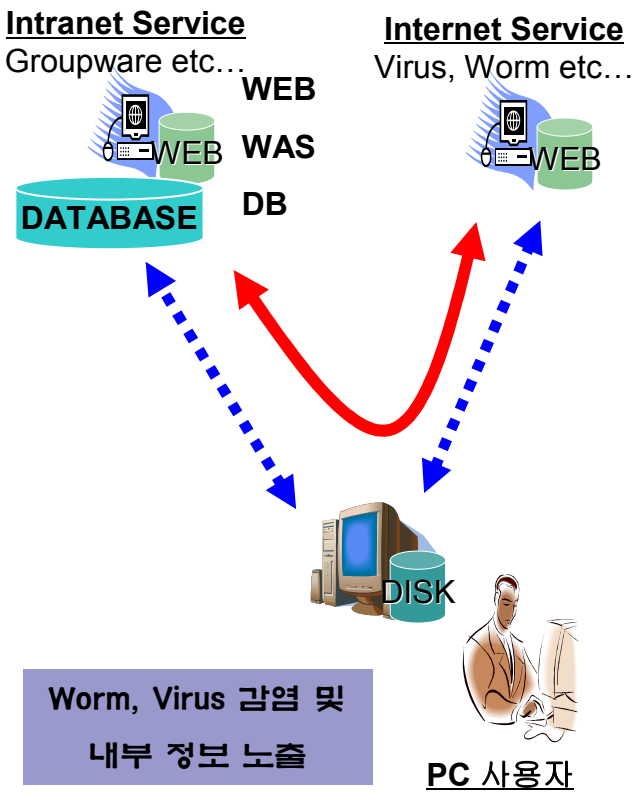


# VM VAC Architecture



Internet 환경과 Intranet 환경의 완벽한 분리를 가상화 기술을 통해 구현하며, 가상화 기술을 통해 논리적으로 인터넷 환경과 업무 환경을 분리하여, EA 환경에서의 획기적 보안성을 강화 시킬 수 있다.

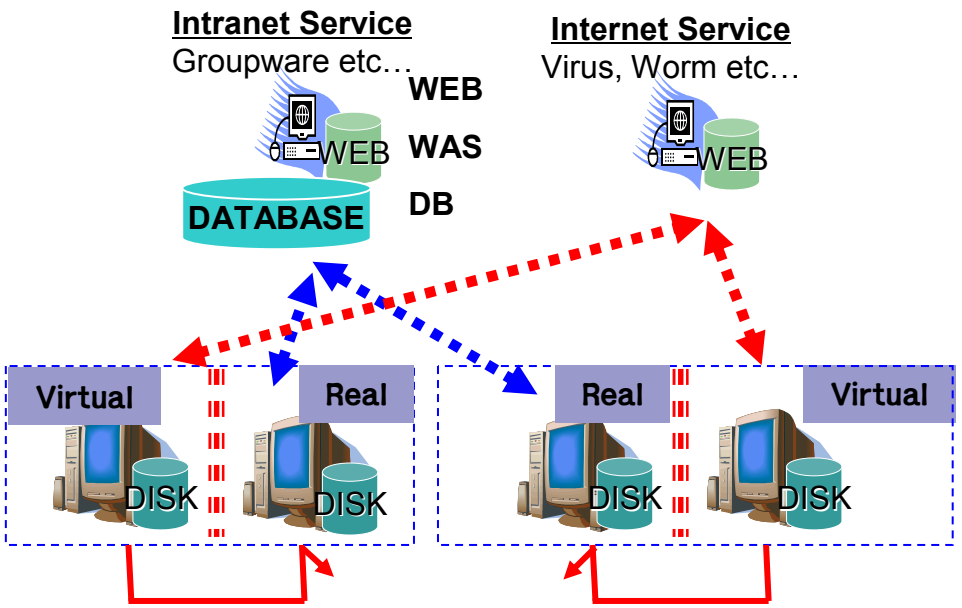
## 기존 접속 **방식**




**Worm, Virus 감염 및 내부 정보 노출**

**PC 사용자**

## 가상화 분리 접속 방식

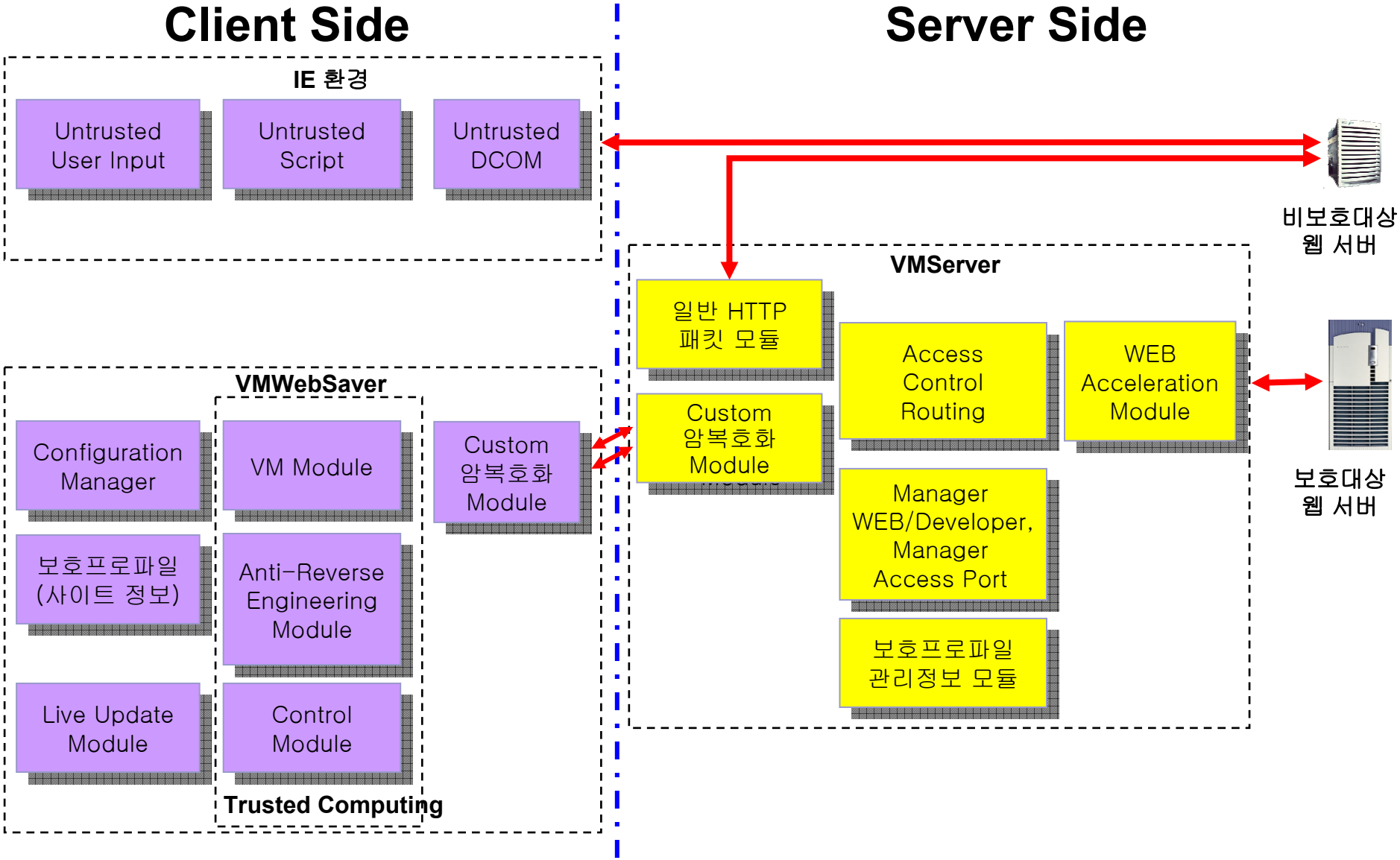



**PC 사용자**

업무 공간과 인터넷 공간은 가상화 기술을 통해 구분되며, 가상화 업무 공간에서만 자료의 상호 교환과 유통이 접근제어 하에 가능하도록 가상화 구현

# VMWebSaver Architecture

www.vmcraft.com



# Virtualization is new real



| 공격 유형 | 방어가능 | 구현 내역 | 비고 |
|---|---|---|---|
| Process/COM/DCOM/COM+<br>Debugging/Reverse Engineering 공격 방어 | O | 가상화를 통해 Real 공간에서 가상화 공간으로의 침입을 방어하여, 공격자의 분석과 공격을 차단 | |
| 암호화 해독 방지(DOM Inspection 방지) | O | DOM 보안 기능을 가상화 기능에 추가되어, DOM Inspection의 방어 | |
| API Hooking 공격 방지(DLL Injection 계열) | O | Real 공간에서 Virtualization에 침입 할 수 없도록 보안 수준 설정에 따라 악의적 DLL 등은 거부 됨 | |
| SQL Injection 공격 방어 | O | 네트워크 속도저하가 극소화 되며, 가상화 클라이언트의 공격 탐지 알고리즘을 통해 공격시도가 이루어질 경우, 네트워크 차단 등의 강력한 대응이 가능 | |
| XSS 공격 방어 | O | | |
| ActiveX 컨트롤 취약점 공격 | O | ActiveX Control에 대한 접속제어를 가상화 I/O에서 구현하여 취약점 방어 | |
| COM 후킹방식 키로거 방어 | O | 공격자는 가상화 시스템을 침입하지 못할 경우, 키로거를 통한 사용자 정보 추출은 불가능 | |
| Proxy을 통한 인자값 위변조 공격 | O | 가상화 공간에서 Proxy를 삽입할 수 업도록 보안기술을 가상화 공간에 구현. | |
| Session hijacking 공격 | O | VMAppSaver와 VMServer와의 통신 구간은 암호화되며, Session Hihacking 공격은 불가능 | |
| Regmon, filemon에 대한 방어 대책 | O | 어플리케이션에 대한 악의적 분석을 가상화 공간의 어플리케이션에 대해 못하도록 구현 | |


Q & A
감사합니다
Han, Seung Hoon / CTO
VMCraft, Inc.
jerry@vmcraft.com