computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

REAL-WORLD ACCESS CONTROL SYSTEMATIC FAILURES; REALITY OR VIRTUAL REALITY?

Philip Attfield [1)]
Ming-Yuh Huang [2)]

1) President, Northwest Security Institute (NWSI), Seattle, USA, attfield@att.net
2) Associate Technical Fellow, The Boeing Company, Seattle, USA, ming-yuh.huang@boeing.com

Abstract: *This paper examines the true causes of systematic failures of real-world access control within the context of modern business transactions. Today's business transactions depend heavily on systems that were developed and protected by off-the-shelf, checklist-mentality security technologies/products such as firewalls, intrusion detection systems and anti-virus software. This dependency, as well as the oversight of system level security requirements, frequently leads to incorrect and incomplete security implementation at the business process and transaction levels.*
To fully illustrate the critical issues faced by today's system, this paper utilizes a real-life cyber crime case for analytical purposes. This case was successfully prosecuted by a jury trial at the *US Federal Court in Seattle during the period of 1999-2000. It revealed many fatal system security failures and business process trust collapses in an environment involving multiple online web-based systems. The paper then shows how such failures are directly attributed from the inappropriate application of technologies/products based on false assumptions of trust, as well as the lack of appropriate security engineering process during the systems development phase. Observations and recommendations are also made regarding what can be done to enhance security and trust requirements at the levels of business transactions and processes.*

Keywords: *case study, business process, transaction, prosecution, cyber crime, systematic security design methodology.*

1. INTRODUCTION

Modern day system development has become increasingly complex and this has led to the common approach of relying heavily on the integration of "off-the-shelf" components.

When systems are constructed in this manner, security functionalities, if addressed at all, also frequently end up being simply "off-the-shelf" component-based solutions. This "checklist-mentality" treats security as a second-class citizen and defines it merely as the sum of security functionalities of all products to be integrated into the system. For the purpose of expediency and convenience, security is usually not tightly integrated into overall system architecture from the start. Two critical questions continue to be ignored. – Is the sum of parts equal to the total? Can security be simply treated as a last minute add-on or should it be part of the entire solution and thus be integrated into the system engineering process?

This paper explores the security implications that result from this prevalent "box integration" and "off-the-shelf" integration approach. The paper will demonstrate that although security components such as firewalls and intrusion detection systems (IDS) may be effective in safeguarding building block system components, they are grossly inadequate for safeguarding the business processes that these systems intend to implement. The sum of the parts is not equal to the whole.

This paper further asserts that "box integration" methodology and "off-the-shelf" security applications make significant, *unfounded* assumptions about the business processes that they attempt to implement. The paper will examine these assumptions and propose a more effective approach to address modern system security requirements. In this approach, security is evaluated, designed and implemented within the context of higher, more abstract layers within the system (i.e., business process) where suitable visibility and knowledge of

the processes and transactions they are intended to protect is available.

Using a real-life case analysis, this paper demonstrates by example where significant business systems, built using "box integration" methodology and "off-the-shelf" security applications, have failed and how the violations went undetected by the system operators. In this particular case, the system failures resulted in millions of dollars of damages and untold numbers of identity theft cases.

2. WEB BROWSER AND WEB SERVER PERSPECTIVES – OPEN SYSTEMS ASSUMPTIONS

Beneath the reality of on-line business transactions, all tangible communication protocols, including HTTP and HTTPS, represent nothing more than a stream of bits formatted according to a specification.

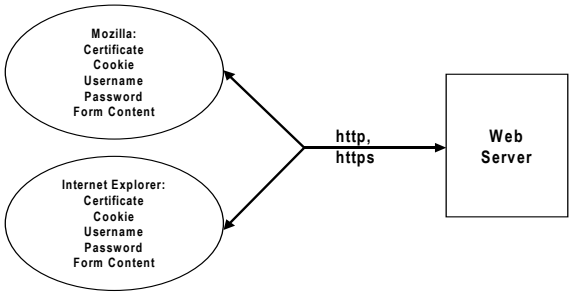

Fig.1 Web Server Perspective of Web Browsers

Most protocols, including HTTP and HTTPS, were specifically designed to permit interoperability between the different components that implement the same protocols. These are known as the "open" systems. From the web server's perspective, the web browser is nothing more than the bit stream it produces. The web server cannot distinguish between different browser implementations provided that the protocol-based bit streams generated are equivalent. Fig.1 illustrates the reality from the web server's perspective.

A similar case may also be stated from the web browser's perspective: web servers that generate identical protocol bit streams are indistinguishable from one another. Indeed, multiple implementations of web browsers and servers co-exist and most of them work happily together most of the time.

Two common assumptions are being made here. Anything that communicates like a web server is most likely a web server. Likewise, anything that communicates like a web browser is likely a web browser. These assumptions carry a significant implication on how a system is assembled in the first place.

3. STATE – WEB SESSIONS

Both web servers and browsers incorporate means of implementing "states" to implement the concept of "web sessions" in support of higher level "business transaction sessions". "Web sessions", for example, can be achieved through the use of cookies or by identifiers that are generated on the fly and embedded in script code for a given session.

"State" is just that, something to be maintained, something to be stored. Browsers frequently maintain cookies and page content within a cache. How state is maintained is irrelevant to the protocol and is invisible to the web-peers (browser or server). However, how the state is maintained has significant security implications particularly at the higher, more abstract levels of business processes. There is a requirement for a trusted *representation* and *processing* of state for a transaction to take place.

Thus, a common assumption being made here is that anything that maintains a correct, consistent, & logical state-based transaction is assumed to be a truthful and legitimate transaction partner.

4. HTTP AND HTTPS SECURITY

Security has been incorporated in several forms within HTTP and HTTPS environments such as X.509, SSL, TLS and PKCS11. All of these are security bolt-ons; largely afterthoughts to HTTP.

X.509, SSL, TLS and PKCS11 represent protocol specifications and all of them come in various implementations. The specific storage, handling, implementation and governing policy of security elements such as cryptographic keys, cryptographic methods and encryption ciphers are only *known* where they are implemented. Implementation details cannot be known, verified or *trusted* by the peer.

If a web server sees a bunch of bits on the wire that appear trustworthy (for example, encapsulated within SSL), then it will, in the absence of other knowledge, unknowingly treat them as trustworthy. This does not mean that the web client is trustworthy; it only means that it generated a sequence of bits that the server *blindly* trusts.

The applications conveniently and falsely assume that secure communication implies trusted identities and legitimate transactions.

5. VIRTUAL WEB BROWSER

Web servers, or rather the designers of web-based systems, assume that web browsers are human-driven at the other end of the wire. This is another unfounded assumption since the web server has no *trustworthy* means of verifying that there really is a human in front of a browser.

If a browser correctly implements the HTTP, HTTPS (etc.) protocol specifications and generates the appropriate security bits then the server cannot distinguish between bits resulting from human interaction and those synthesized entirely in the absence of a human, i.e., automated, or scripted. This represents a significant security and trust gap in web-based systems where the intent is to provide secure, trustworthy services to humans. When an on-line transaction system is designed, such security considerations should be taken into account from the start. In the "box-integration" approach, such considerations are often neglected and never considered.

Therefore, a virtual web browser can be built to implement HTTP and HTTPS protocols, maintain state, and incorporate security features such as X.509, SSL and TLS. It is also possible for such a browser to interpret the bit stream received from a web server and synthesize user actions such as filling in a form or pressing a button.

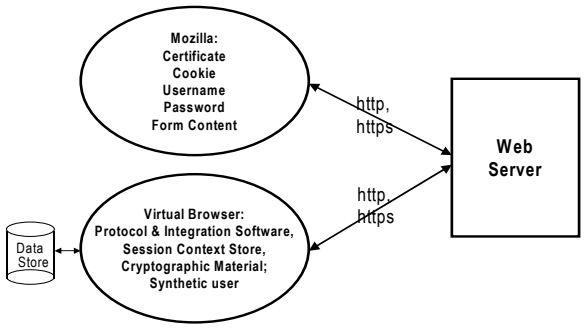

Fig.2 Virtual and Human Web Browsers; Web Server Perspective

Fig.2 illustrates the situation of a web browser receiving traffic from both a human-driven web browser and a virtual browser. Traffic from the two sources is indistinguishable.

6. SYNTHETIC USERS

Web proxies and relays have been developed as a means of bridging between routable and un-routable Internet Protocol (IP) address spaces as well as providing firewall capability. Web proxies and relays can both be constructed to mask the true origin of the web traffic. The web server's view of the web client is thus further obscured. These intermediate, "apparent traffic" origins can also be used by criminal elements to mask routes tracing back to sources of undesirable activity. The usage of web proxies and relays can further amplify the deployment and effectiveness of "virtual web browsers", permitting a single virtual browser backend to mimic the behaviour of a large number of "human" users. If web proxies and relays are suitably distributed across the network, then traffic from the originating backend blends very well with traffic from legitimate human-based browser activities. Under such circumstances, activities generated from the "virtual browser synthetic users" become almost undetectable. Detection attempts can be further hindered if the intermediate relays/proxies are chosen from a large pool of widely distributed systems.

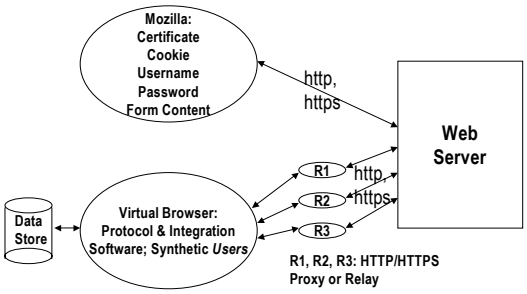

Fig.3 Web Server traffic from Human-Driven Browser and Virtual Browser Synthetic Users

Fig.3 illustrates a scenario where a virtual browser backend utilizing web-proxy/relays effectively mimics human users. Other network protocols such as DHCP (Dynamic Host Control Protocol), that manage the assignment of IP addresses, can further exacerbate the problem of differentiating between synthetic and human users.

7. VIRTUAL MESSAGING

Numerous "free" web-based email services exist. These include for example, Yahoo!, Gmail and Hotmail to name a few.

A feature-rich messaging system can be constructed using the virtual web browser and "free" email services. Such a messaging system can:

- Create and manipulate web-mail accounts

- Send, receive, parse and otherwise process messages
- Utilize a database system to maintain *user context,* as well as message context, message content, web session and business transaction states.

Use of intermediate web proxies/relays allows the messaging system to appear as multiple "human" users. Traffic and *user-activity* resulting from the synthetic users easily goes undetected by the email service provider.

One fatal assumption is being made here -- a logical sequence of email messages validates the legality and trustworthiness of an on-line transaction.

8. VIRTUAL PAYMENT TRANSACTIONS

There are numerous web-payment services in existence today. These services frequently associate a bank account or credit card to an internet identity. Their intent is to facilitate the transfer of funds between parties conducting business online. The "PayPal" online payment system is an example.

The virtual browser can be used to create an automated payment system. It can incorporate features that:

- Create and manipulate accounts
- Associate personal information with the account such as credit card number, bank account and branch information, billing address, mailing address, and *email* address
- Address the transfer of money
- Limit money transfers to less than the $500 threshold thereby defeating other security measures
- Work in conjunction with multiple web proxy/relay systems to create a large number of synthetic *users*
- Work with a backend database coupled with a messaging subsystem to emulate these synthetic users

With an appropriately constructed virtual browser, traffic generated by these synthetic users is again indistinguishable from human users.

9. VIRTUAL PAYMENT TRIGGER MECHANISM

A virtual browser, along with the synthetic users that it implements, can be used to trigger payment transactions.

Ebay is a very well known and successful web auction service. A seller places items for sale. These items are then subjected to bids from prospective purchasers. Both the seller and purchasers are registered with the auction service. The prospective purchaser who places the highest bid "wins" & "buys" the item when the auction concludes. The conclusion of the auction triggers a payment transaction between the seller and buyer.

A virtual browser can be constructed for fake auction services. It can include features that:

- Create and manage accounts (for both buyers and sellers)
- Associate email addresses and physical delivery addresses with the account, maintained and managed by a database.

The virtual browser now represents typical users of the web system. If the virtual browser also has the ability to:

- Create auctions
- Add bidders
- Automate bidding
- Automate payment at auction close, and
- Automate buyer and seller feedback

Then, the virtual browser, combined with proxy/relay intermediates that give the appearance of multiple, legitimate, synthetic users, can literally create, manage and *pay* for items that do not even need to exist.

10. VIRTUAL BUSINESS REALITY

The user enrollment procedures of web services such as email, payment and auctions are based on users' *goodwill* rather than founded on trust or verifiable security models. The basic assumption of the individuals who designed these systems is that users will be well behaved.

In reality, this is a weak basis upon which a system designer should make assumptions regarding the trustworthiness of end users. Systems based on these assumptions will fail if subjected to web traffic created by synthetic users.

If the capabilities of the email, payment and transaction processing are combined, then a system that automates fraudulent money laundering is created. The participating web service providers would not be aware of their involvement in the scheme beyond seeing an increase in network traffic and system activity. There is very little that today's

IDS (Intrusion Detection Systems) can do to help. The payment services provider might notice an increase in fraud, but fraud is part of their business cost model and is often covered by insurance or contract clauses. Fig.4 illustrates such a combined "virtual" business.

Given the potential of the significant quantities of money that can be laundered in this way, it should come as no surprise that such an incident has already happened. During 1999-2000 Ebay, PayPal (still a separate company from Ebay at that time) and several web-email providers were involved in just such a crime.

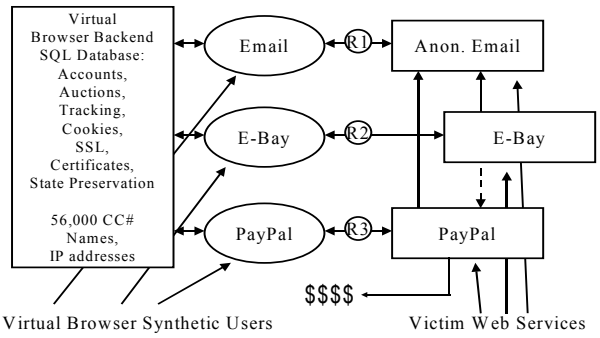

Fig.4 Virtual Browser: Virtual Business Reality

The criminal activity was prosecuted successfully by the Federal Bureau of Investigation of the U.S. Department of Justice in September 2001 in the case United States v. Gorshkov.

11. CRIMINAL CASE AND ITS EVOLUTION

The evolution of the crime, law enforcement participation and case prosecution provide useful insight regarding how complex system failures occur and how they can be analyzed.

The criminal activity took place during 1999 and 2000. Several complaints were filed with the F.B.I. in multiple jurisdictions including computer intrusion, system outage and attempted extortion. The coordinated effort of several offices and investigating agents ultimately resulted in an undercover operation that took place during November 2000.

Two suspects were arrested in Seattle as a result of the FBI undercover operation that involved fictitious international job advertisements and interview offerings. They were subsequently charged with numerous offences.

The internet-connected computers at the undercover operation were fitted with keystroke recorders. One of the suspects logged in to their "home system" and the keystroke recorder obtained the system name, username and login password. The F.B.I. subsequently reconnected to the remote system and downloaded approximately 2.3GB of compressed data.

The downloaded data was analyzed in conjunction with data obtained from victims' systems. This revealed the true nature and the extent of the criminal activities that had been conducted.

Seized evidence and victim data revealed that the following types of incidents took place during the 1999-2000 time frame:

- Numerous computer intrusions including the subversion of systems and networks, for example ATM connected systems at a school district in Michigan
- Computer outage, for example at an internet service provider in Bellevue, Washington
- Credit card fraud, for example at online retailers and internet payment systems
- Attempted extortion, for example at a bank in Southern California.
- Large-scale identity theft.

Compromised systems were frequently used as web relays/proxies. If the compromised system had "business value" then it was also used for other purposes. In one instance a system connected to a high-bandwidth ATM network was employed as a Domain Name Server (DNS) and Internet Relay Chat (IRC) server. In another instance the web site of an online bank had undergone creative enhancements that bypassed the normal user log-on procedure.

The evidence also contained numerous Perl programming language software scripts and temporary file residuals resulting from their execution. The Perl scripts implemented a virtual web browser and were customized for email, auction and payment functions.

The Perl scripts appeared in numerous forms of developmental evolution ranging from simple connection test scripts, SSL connection test scripts with embedded links to X.509 certificates through to connectivity to a fully integrated backend database.

Thousands of email addresses were mined from the seized evidence. These addresses were correlated to activity at a web email service provider.

The Ebay auction scripts represented a full-function user account creation/management and auction creation/bid/close capability. The auction management capability also included a feature that limited transactions to below the $500 PayPal threshold. Support for the automated generation of Ebay buyer and seller feedback was also incorporated. The $500 threshold check and automated feedback represent a deliberate "fly below the radar" strategy.

The PayPal scripts demonstrated the capability of being able to create and manipulate PayPal accounts. The PayPal accounts were associated with stolen credit card information.

The ingenuity of the Perl scripts also provided clear evidence of the suspects' in-depth understanding of the operation of the email, PayPal and Ebay web servers. All of these systems had been carefully analyzed and effectively reverse engineered by the suspects. Not only were the systems understood in terms of implementation technology but also in terms of business level transactions and the relation between these systems and others with which they had interaction.

The nature of the computer intrusion attack methodology is also noteworthy. We frequently hear that attacks follow an intelligence and reconnaissance phase. Intrusion Detection Systems typically report sensor events during reconnaissance probes (e.g., port scans).

The intelligence phase of these attacks consisted of assembling a long list of potential target systems. *There was no need to initiate reconnaissance probes.* One-click "attack/compromise/subvert" scripts were developed. These scripts targeted known and "as-yet unknown" (i.e., unpublished) vulnerabilities and fully automated the installation of trojan-horse software, root kits, web proxies/relays as well as the search, gathering and retrieval of information contained on the compromised system and the network to which it was attached to. The targeting of the "as-yet unknown" vulnerabilities highlights the limitations of today's IDS and anti-virus systems which are primarily based on the "20-20 hindsight" band-aid approach and working at the wrong level of abstraction. It also highlights the fallacy that there is benefit in keeping unpublished security vulnerabilities secret until patches are available.

12. CASE DEVELOPMENT AND OBSERVATIONS

The analysis of seized evidence and victim data was a time consuming activity that spanned approximately one year.

A methodology is involved in the digital forensic engineering process for evidence analysis, scenario development, evidence re-construction and case building. First, evidence analysis led to the creation of a hypothetical system model that accounted for seized evidence, victim evidence, and victim observed activity and event times.

The model was subjected to continuous validation such as "Can the presence of a backend database be substantiated?", "Can Ebay verify that feedback generated by the Ebay virtual browser Perl script exists within their systems?", "Can the email provider verify the addresses that have been mined from evidence?". These validations helped to confirm the model as well as further identify and reveal the extent of the criminal activity.

The model, seized data and victim related activity or knowledge led to the following observations:

- Numerous web service operators were not aware of their participation in the crime
- The free email service providers remarked that they had experienced heavy traffic but were otherwise also unaware of their involvement in the crime.
- Web service user enrollment (i.e. subscriber verification) and identity verification procedures were revealed to be inadequate.
- The use of unidirectional SSL, whereby the server blindly trusts the client, allowed the virtual browser to assume multiple, false, unverified identities.
- User certificates, if generated, were not "bound" to a trusted device by a trustworthy enrollment process. The virtual browser employed whatever technology necessary (i.e. X.509 certificates, SSL and bi-directional SSL) to masquerade as an undetectable, *synthetic* user.
- PayPal was continuously subjected to substantial fraud losses. Since the majority of all of their payment activity was triggered by Ebay auction transactions, and they blindly trusted that all the Ebay transactions they received were valid, these transactions would probably have been viewed like all others – 'as a good thing'.

- Ebay noted that traffic was heavy but otherwise they were unaware of the fraudulent auctions that were taking place. They were shocked when they learned that even the seller and buyer "feedback" auction features had been exploited.
- Ebay participants naively believe that "feedback" is a measure of trust. Subversion of the feedback mechanism by the generation of automated fake buyer and seller "feedback" entries underscores the point that feedback is solely based upon Ebay participant goodwill. There was absolutely nothing trustworthy about Ebay feedback.

One of the suspects was subsequently brought to trial in Seattle in September 2000.

It should be noted that, during the court proceedings, the model that was developed to account for the seized evidence, victim data and criminal activities was presented and accepted as a major part of the total body of evidence. Other facts revealed during the trial include:

- Approximately 56,000 stolen credit cards (identities) were involved
- An unknown, significant number of proxy/relay systems were deployed
- Damages in excess of $USD 25,000,000 were inflicted

The accused was ultimately convicted by the jury on all twenty charges and subsequently sentenced to jail time.

13. CONCLUSIONS

Many assumptions are being made in modern large-scale system construction. Most are simply the result of natural technology evolution and some are inherited from the old computing paradigms. These assumptions are highly implicit and very much obscured to the system owners.

Following the old model of "security as a last minute add-on", people build systems, discover security vulnerabilities and, only then, patch them. Assumptions are conveniently *assumed* to be true until they are proven to be false. However, as illustrated by the case in this paper, many of the old assumptions are simply breaking apart. The reality is that systems constructed using tools and building blocks does not necessarily equate to a well-built and secure system. There is an urgent need to carefully examine many critical system design and construction issues from a more systematic approach. Without this, next-generation business transaction models (e.g. virtual enterprise and on-line banking/auctions) will remain very much at risk.

Sensors are limited in their visibility. Businesses that provide web services such as email, payment and auction have severely limited abilities to detect their users' participation in such illegal activities because their ability to observe is well constricted within their own domains. This constraint holds true in both computing and business contexts. Not having the sensors in the right place or not sampling data at the right time within an on-line business transaction system guarantees that unusual behavior will go undetected. In fact, improper placement of sensors can convey the false impression that everything is normal and "safe". The problem is exacerbated for highly distributed and complex systems. This mismatch of detection and application architecture is a direct result of the "box-integration" approach.

Sensors and security applications are at the wrong level of abstraction. As the above case clearly illustrated, neither today's neither network-based nor host-based intrusion detection systems can be of much value in this kind of real-life transaction-level intrusion. The old paradigm (assumption) of securing the platforms (network, operating system) to secure the content (data, transaction) is failing. Consequently, sensors suitable for the platforms (network and host-based IDS) are not necessarily appropriate for detection at the application and transaction levels.

Many off-the-shelf security applications are designed to address security problems within the network protocols, network stack and operating systems (buffer overflow, file system, virus). However, the security of the server is a totally different matter from the security of the transactions or business processes that the server is intended to support. Off-the-shelf security applications do not have any visibility or knowledge of business processes and transactions. They were never intended to, and are unable to, provide protection at this level. Therefore assumptions regarding their ability to do so are simply not valid. Trust is assumed where it should not have been and as a consequence business processes are left vulnerable and prone to failure. This absence of security is a critical design failure. Systems constructed in this manner can be completely subverted without being compromised. They can also place their operators within a criminal context.

What is not observed can never be seen. If a *business* process is not *monitored* at a level of abstraction corresponding to the abstraction level of the *business process* then all activity at that level, *especially the unusual*, goes undetected. This principle is well understood within the banking community where systems incorporate transaction models, audit logs and monitors.

System development methodology must incorporate security as part of overall system engineering process from the very beginning of the system design. The security engineering process must be able to clearly capture the requirements, design, implementation and maintenance of the system. In such a manner, the process explicitly addresses requirements such as adequate observables and monitoring of business processes for transactions that relate to business process trustworthiness. The process must also specify the transaction trust issues and risk management processes. Protection of the server, operating system or file-system should form part of the security requirements, but is not adequate by itself given the complexities of today's business processes.

Algorithms used within off-the-shelf IDS systems might be entirely appropriate for use at the business process and transaction levels provided that they could be modified to understand abstract business process and transaction models. Such modifications would be business process specific and therefore not conform to the "security-checklist mentality" design methodology. It would also require system designers to develop rich business process and transaction models of their systems. These models would need to incorporate high-level representations of threat scenarios.

Business processes, especially those relating to user enrollment procedures, need to be seriously considered when designing systems that are intended to be used solely by humans. Automated fraud and money laundering systems have been proven feasible.

14. REFERENCES

[1] Court proceedings and public-record trial exhibits, United States v. Vasily Gorshkov; Seattle, WA, U.S.A, September 2001.

[2] Philip Attfield, Expert Witness Analysis and Testimony, United States v. Vasily Gorshkov, Seattle, WA, U.S.A., September 2001.

[3] United States Department of Justice press release, "Russian Computer Hacker Convicted by Jury", Seattle, WA, U.S.A., Oct. 10, 2001.

[4] United States Department of Justice press release, "Russian Computer Hacker Sentenced to Three Years in Prison", Seattle, WA, U.S.A., October 4, 2002.

[5] United States Department of Justice press release, "Russian Man Sentenced for Hacking into Computers in the United States", New Haven, CT, U.S.A., July 25, 2003.

Philip Attfield *is currently the President of the Northwest Security Institute (NWSI), Seattle, USA. He is also an active board member and adviser to IDELIX Software Inc., a technology company based in Vancouver, Canada. He continues to contribute to activities stemming from his role as facilitator at G-8 Cybercrime conference data retention workshops.*

Mr. Attfield previously worked at Boeing within their Phantom Works R&D unit where he led a team of researchers engaged in the development of a scalable, policy authorization framework. He was employed by the U.S. Department of Justice during his Boeing tenure and worked with FBI investigators as well as DOJ prosecutors on "U.S. v Gorhskov", where he testified as expert witness.

He was also the founder and CEO of Signal 9 Solutions, a pioneer in Common Criteria certified desktop/personal firewall software. Signal 9 Solutions was acquired by McAfee in 2000.

Mr. Attfield holds M.Sc.E. and B.Sc.E., both in Electrical Engineering from Queen's University at Kingston.

Mr. Ming-Yuh Huang *is a Boeing Associate Technical Fellow responsible for leading Boeing's corporate Information Assurance R&D program that supports many large-scale commercial/military programs as well Boeing's own, one of world's largest, corporate infrastructures. Before joining Boeing in 1990, Mr. Huang worked at DEC Artificial Intelligence Technology R&D Center in Boston as a Principal Researcher leading DEC's intrusion detection expert-system research effort ESSENSE (Expert System for Service Network Security). ESSENSE resulted in one of world's first commercial intrusion detection product. Mr. Huang is recognized internationally as a pioneer of IDS (intrusion detection system) technology and a leader in the IT security. While with Boeing, Mr. Huang led DARPA intrusion detection research project, co-chaired premier international IDS conference RAID-1999 (International Symposium on Recent Advances*

in Intrusion Detection) at Purdue University, and was twice invited by EC (European Commission) to participate in defining EU/USA IT security R&D collaboration framework. Beside numerous conference/workshop publications and journal editorship, he also co-authored IETF (Internet Engineering Task Force) IDS communication protocol standard in collaboration with IBM R&D Lab and AFIWC (US Air Force Information Warfare Center). Mr. Huang has served as the keynote speaker for many European and Middle East IT security conferences. He was the Program Chair of 2004 NATO ARW (Advanced Research Workshop) "Cyberspace Security and Defense: Research Issues" in Gdansk, Poland - http://kio.eti.pg.gda.pl/arw2004/. He is also the General Chair of RAID-2005 in Seattle, Washington - http://www.conjungi.com/RAID/ and Program Chair of SADFE international digital forensic workshop in Taipei, Taiwan - http://conf.ncku.edu.tw/sadfe/index.htm. Mr. Huang has a B.S. in Physics, a M.S. in Computer Science, and (incomplete) study at University of Oregon Computer Science Ph.D. program.