

the security journal

**SECURITY... CAN BE
BROKEN, WILL BE BROKEN.**

WHAT WILL YOU DO?

WHAT WILL YOU DO?

***IN THIS EXCITING
EPISODE...***

The government secures my
information....right?

.....

Think safety, *before* touching
that keyboard!

.....

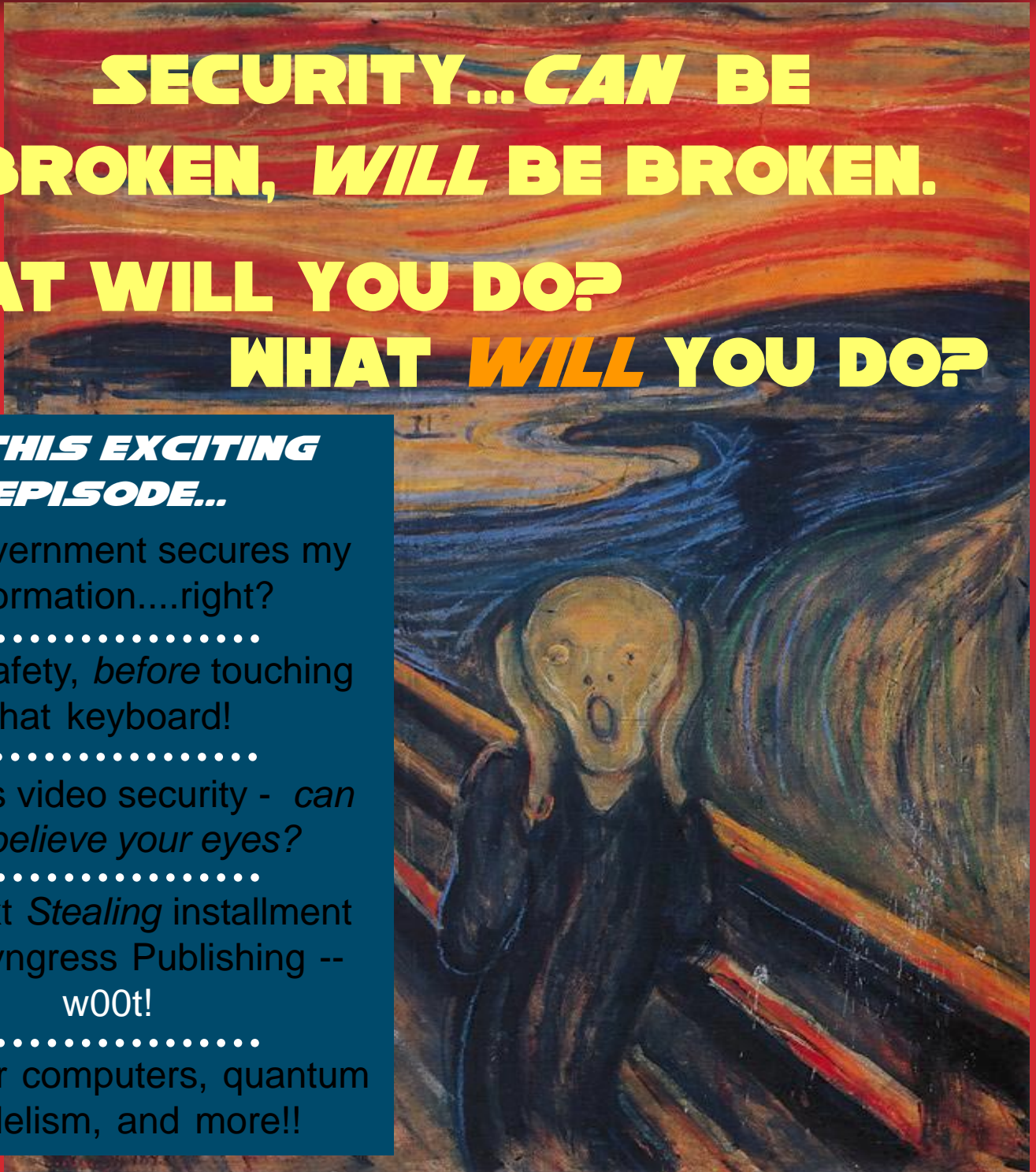
Wireless video security - *can
you believe your eyes?*

.....

The next *Stealing* installment
from Syngress Publishing --
w00t!

.....

Molecular computers, quantum
parallelism, and more!!





Your global information security experts

**5350 Tomah Drive
Suite 3500
Colorado Springs, CO
80918**

<http://www.securityhorizon.com>

Editor in Chief: Russ Rogers
Assistant Editor: Michele Fincher

The Security Journal is always looking for quality writers for this publication. If you're interested in submitting an article for an upcoming edition, please contact us at editor@securityhorizon.com.

Interested in advertising in The Security Journal? Contact us at editor@securityhorizon.com

Special thanks go out to Ping Look from **Look Designs** for the creation of the Security Horizon logo and masthead images for *The Security Journal*.

For more information on Look Designs:
www.republicofping.com
design@republicofping.com

For more information on Security Horizon, please visit our web site or email us at: info@securityhorizon.com.

Security Horizon, Inc. was founded in 2000 and is headquartered in Colorado Springs, Colorado. As a company, we believe that sound security services are based on education, experience, standards, ethics and unquestionable integrity.

Table of Contents:

Notes from the Editor
By Russ Rogers.....p. 3

Secure Programming Concepts
By Scott White.....p. 5

NSA IAM Course Schedule
.....p. 7

The Security of Homophonic Data Encryption Standard
By Geetha Sivakumar.....p. 8

NSA IEM Course Schedule
.....p. 11

Book Excerpt
By Syngress Publishing.....p. 13

Wireless Video Security
By Russ Rogers.....p. 26

Electronic Signatures and Public Key Infrastructure, Part 2
By David Sweigert.....p. 30

Brian's Tool Corner
By Brian Kirouac.....p. 32



Cover credit: *The Scream* by Edvard Munch

Copyright ©2006 by Security Horizon, Inc.
All rights reserved. Reproduction without permission is prohibited.



Black Hat[®] **Japan 2006**

5-6 October 2006

Keio Plaza Hotel, Shinjuku, Tokyo

Briefings: 2 Days, 2 Tracks

IA **japan**

On 5-6 October 2006, Black Hat together with Internet Association Japan
will host the third annual Black Hat Briefings Japan in Tokyo.



Call for Papers and Registration now open.
japan.blackhat.com

Notes From the Editor



Russ Rogers
CISSP
Editor in Chief



Building Trust: Revisiting Our Processes

As information security professionals, we tend to spend an inordinate amount of time trying to cajole and convince our co-workers, customers, family, and friends to secure their important and sensitive information. We do that because it's what we do, and if you're anything at all like me, it's a mission; something that you tend to focus on. Even in mundane situations, it sometimes appears as if I'm being a little bit too paranoid. But recently, it almost appears as if the industry has grown lax in its duties. Maybe it's just that these things become somewhat larger than life when they're picked up by the press, but we're making some fairly big mistakes these days.

To narrow this down a bit, I'm talking about the apparent outbreak of lost, stolen, or misplaced information from some major organizations over the last 2 months. There

have been at least three very noteworthy incidents, in the last 2 months alone. All of these examples concerned security breaches where a lot of sensitive information has leaked out beyond the organization's control.

Let's start with Washington, D.C.; hub of the country and proud home to our National leadership. Somewhere around June 19th, 2006, a laptop containing private and sensitive information on roughly 13,000 Washington D.C. employees and retirees came up missing, along with personal information including the names and social security numbers of those individuals. The laptop was stolen from the home of an ING Financial Services employee's home in Washington D.C. ING administers the District's retirement plan. The laptop was not password protected and the information about the employees was not encrypted. The sad truth is that there are now 13,000 more people out there that are susceptible to identity theft, fraud, and other crimes.

Our second incident comes to us from Hotels.com. In May of 2006, I received a letter from Hotels.com stating that my personal information may have been on a laptop that was stolen from the company's auditor. The auditor in question, Ernst & Young, had the laptop stolen from an employee's locked car. The laptop was stored in a backpack and required a password to get into. However, the article does not state whether or not the information was encrypted. It simply states that the file was several directories down from the root level and did not contain a filename that would specifically identify the file as having customer information.

If we stop to consider the facts here, we find ourselves scratching our heads. How does Ernst & Young, a leader in the auditing world

Notes, cont.

and birthplace to security experts such as the founders of Foundstone, find itself in a position where it can not secure its customer's sensitive information? As a result, "some customer transactions . . . primarily from the year 2004", may have been compromised. This includes names, addresses, and credit or debit card information. One report estimates roughly 243,000 customers potentially affected by this incident.

The biggie wasn't either of these events, unfortunately. On May 22nd, 2006, the VA announced that a data analyst from the VA had taken home a Department owned laptop, which was directly against organizational policy. The laptop was subsequently stolen from this employee's home. On this laptop was the information on some 26.5 million veterans and their spouses; including the names, social security numbers, birthdates, and disability ratings of those veterans. No information was given regarding whether the information was encrypted, the laptop was password protected, or whether any mechanism to deny access to the data was in place at the time the laptop was stolen.

As a side note, just this morning, the VA announced that the laptop in question had been turned into the FBI in Baltimore. After preliminary investigation (it doesn't say what sort of investigation, but I'm assuming forensics of some sort), they believe that the information on the drive was intact and had never been copied or modified. It will be interesting to see what the final outcome is on this particular incident.

But when we look at news like this, it's easier to understand why some customers are worried about hiring an outside security firm. Based solely on the mass media, we don't have a great reputation. And who in their right

mind would want to be the CEO of a Fortune 500 organization that happened to hire the security team that lost thousands of customer records? It costs a nice chunk of change to bring a professional into the organization and the last thing we want is our name plastered all over the news. And it's obvious that the higher priced security organizations are no better off than the smaller companies.

But there is a solution, a fix, to the problem. We need to revisit our processes and reeducate our security personnel. The processes we use internally should be more robust and stronger than those we recommend to our customers. Password protecting laptops, encrypting all sensitive information on the hard drive, and securing handling laptops and computers are all sound policies. Hotels.com has responsibility for the security of all of their customers. But as the security consultant or trusted auditor, you have responsibility for all of your customer's customers. This is a large part of why the ISSA and ISC(2) organizations require members to acknowledge a Code of Ethics.

I recently heard a complaint from a security consultant in one of my classes about how customers simply aren't willing to fork out the big check to pay for their services. They said, "I don't think they (the customer) are taking this seriously enough." My response to that is, "I'm not sure we're taking this seriously enough." Remember, this isn't all about the paycheck at the end of the day. It's about making the way we all conduct business a little bit safer.

-Russ

We want to hear from you!
Please send your
comments to
editor@securityhorizon.com



.....

There will always be a security trade-off with respect to time, cost, and feasibility...

.....

I've been asked many times if there is a way to completely secure a computer. I like to reply with, "Yes, only if it is unplugged from the wall." Although a computer that is turned off and unplugged completely from a network and power source is secure from being hacked into remotely, it still can be physically stolen. I am a firm believer that there is no such thing as a secure computer. Just like hardware, software falls into the same category. Software can never be completely secure, only have its security improved. As with other types of security implementation methods, it comes with a cost. Software security measures can be taken, but may cause the application to run slower. There will always be a security trade-off with respect to time, cost, and feasibility. This article will introduce basic concepts to take into consideration to make code more secure.

Global Variables – Only use global variables when needed. Global variables are

accessible by all executing code. Sloppily written code may inadvertently modify global variables causing unwanted behavior while executing. There are a few reasons to use a global variable but should be avoided in most cases. One case where a global variable might be used is to store a shared database connection or connection string.

Threading – Exercise caution with multithreaded applications. Threads executing in parallel may cause collisions or corrupt data that may be used by the application itself, but shared among threads. When multiple threads are accessing a shared variable such as a counter, be sure to implement a locking mechanism to avoid a race condition. A race condition is when threads are modifying a variable such that each thread counteracts the previous threads change. For example, the following table illustrates two threads executing in parallel in such a way that may inadvertently cause a

Thread	Operation	Value
main	int count = 0;	0
Thread 1	count++;	1
Thread 2	count--;	0
Thread 1	count++;	1
Thread 2	count--;	0
And so on...		

denial of service attack.

Iteration (Looping) – Exercise care when iterating or enumerating types. Iteration can be especially dangerous when care is not taken when specifying iterator conditions. In a for loop, `for (int i = 0; i <= 10; i ++)` will loop 11 times. The code: `for (int i = 1; i <= 10; i ++)` will loop 10 times. The programmer may only want the loop to run 10 times. When iterating a very large number of times, a simple error in the iterator operator being *less than or equal to* or *less than* may cause the loop to execute the wrong number of times. In certain cases, this could be extremely dangerous. For example, if the code were adding data to a storage area whether it is a database or buffer, an overflow condition may exist.

stack object “on the fly” within multithreaded code. Variables should be declared only in the scope that they need to be in, and not all methods should be publicly available for use. Using the *protected* or *private* reserved words in many languages, not all variables and methods will be inherited by subclasses.

Error Checking – Be sure to ALWAYS check for errors. Make sure you catch all errors that the code you write might give. For example, if you are opening a .NET SqlConnection, the `.Open()` method would be used and may throw an *InvalidOperationException*. That particular exception means that a data source or server was not specified, or the connection is already open. A good practice is to use *try-catch* blocks to handle exceptions.



Scope and Protection Levels – When declaring variables or methods for a class, as with any security, permit access only when absolutely needed. If your code needs a value defined for PI, such as 3.1415926535, use the *static* or *const* keyword to ensure that its value never changes throughout the execution of the code. An example where public scope is ok, but where local scope would get you into trouble, is using the stack in a multithreaded application. It would be acceptable to declare *public static Stack s = new Stack();* whereas instance members are not acceptable. That is, do not declare a new

Input Validation – Lastly and quite possibly the most important, ALWAYS check user input. If the expected input from a web form is an integer, make sure it isn't an alphabetic character, special symbol, or blank. If your code is a shopping cart allowing a maximum order of 99 items, set the *maxlength* property of the textbox you are using to 2 to prevent more than two characters from being entered into the textbox. Regular expressions can be used to strictly enforce input requirements effectively. Be sure to check for blank or null values. Make sure to check for minimum and maximum input lengths for buffer overruns

Secure Programming, cont.

and overflows. No matter how good of a programmer one might think they are, they are too close to their code to thoroughly and properly test it. Quality assurance testing, using users that are unfamiliar with the program or webpage, will reveal more issues than the programmer may find on his/her own. End users will almost always attempt to use software in ways in which it wasn't intended to be used. Extensive testing is of utmost importance when developing software. In all, any data that may be available for user interaction should have special precautions taken.

The items discussed in this article only touch the tip of the iceberg when addressing code security. Buffer underruns, buffer overflows, data injection, data modification, data deletion, data corruption, and many other unwanted actions may occur if proper programming conduct is not followed. The issued of code security should not be taken with a grain of salt. Software programming errors have resulted in everything from deaths from too much radiation to lost luggage at the Denver airport. No matter how insignificant a programmer might think the code is that he or she is writing, security is an extremely important factor to be taken into consideration.



Scott White is a graduate student pursuing his master's degree in network security at the University of Advancing Technology in Tempe, AZ. He holds a BS degree in Computer Science from Ohio Northern University. Scott has been programming in various languages for 9 years. He is currently employed as a web developer writing C#, ASP.NET, and SQL for Fulton Homes in Tempe, AZ.



7/29 - 7/30, 2006	Las Vegas, NV <i>Black Hat</i>
7/31 - 8/1, 2006	Las Vegas, NV <i>Black Hat</i>
8/11 - 8/12, 2006	Omaha, NE
9/26 - 9/27, 2006	Atlanta, GA
10/9 - 10/10, 2006	Knoxville, TN

The IAM is a two-day course for experienced Information Systems Security analysts who conduct, or are interested in conducting, INFOSEC assessments of information systems using NSA's methodology. It is a high-level, non-intrusive process for identifying and correcting security weaknesses in information systems and networks.

FREE 1 Year license to the SAINT Vulnerability Scanner just for attending any of our courses!

**Training so good we teach
the competition!**

To register for one of these courses or to get further information, please contact us at:

(719) 488-4500
info@securityhorizon.com
<http://www.securityhorizon.com>



The Security of Homophonic Data Encryption Standard



Abstract: DES [8] is a block cipher cryptographic algorithm used for encryption/decryption. A homophonic DES [9], a variant of DES was proposed by Tsu-Miin Hsieh, Yi-Shiung Yeh, Yung-Cheng Hsieh, Chan-Chi Wang. This paper describes how HDES withstands attack due to quantum parallelism, differential attack and molecular computers. The security of this cryptosystem lies on the random number generator. Also, we explore the possibility of generating random numbers with a high degree of randomness.

Keywords: DES, Homophonic DES, Random numbers, Quantum computing, Differential attack, Molecular computing.

HDES

A homophonic DES maps a 56 bit plaintext to a 64 bit cipher text using a 56 bit key. Let P be the set of plaintexts of length 56 bits and C the set of cipher texts of length 64 bits. The mapping E_k from P to C adds 8 random bits to 56 bit plaintext and encrypts it using DES. The eight random bits are placed in specific positions of the 64 bit input data block to maximize diffusion. The criteria for embedding are:

1. After the initial permutation, the eight random bits should all be rearranged to the right half of the data block.
2. In the first round, all eight random bits should be duplicated by the expansion permutation
3. In the first round, each S-box should have 2 input bits that come from two distinct random bits.

Based on the above criteria, 6 arrangements are possible. Thus the algorithms are named



HDES1, HDES2, HDES6. The decryption of the Homophonic DES is similar to the decryption of DES. The only difference is that the 8 random bits must be removed to get the original plaintext.

Attacks on DES

Differential attack
Quantum computing
Molecular attack

Differential attack

A well known attack on DES is the differential cryptanalysis [3-5] of Eli Biham and Adi Shamir. Differential attack makes use of the exclusive or difference of plaintext and cipher text pairs. It estimates the probability that certain plaintext difference will result in a certain cipher text difference by estimating the probability of intermediate difference patterns in the DES algorithm. A difference pattern which occurs with probability can be useful for deduction of some key bits. Right pairs which generate the desired difference pattern will suggest some key values including the correct one,

oppositely, wrong pairs suggest random values. For a difference pattern which occurs with high probability the right key values will be suggested with highest frequency, when enough plaintext pairs have been analyzed.

In HDES, the input bits contain eight random bits which are not known to an attacker. This results in the non determination of the input difference for a plaintext pair. In other words, when a pair of plaintexts (56 bits) are encrypted, the attacker does not know the exact difference of their corresponding input bits(64 bits). Actually, there are 256 possible differences. Thus, for a chosen a pair of plaintexts, the probability that they generate the desired difference pattern reduces to about 1/256 of the probability in case that input difference is clear. The differential attack that requires 247 chosen-plaintexts to attack full 16 round original DES now needs 256 times that number of chosen plaintexts to attack HDES i.e., 255 chosen-plaintexts. Hence differential attack on HDES is very difficult.

Quantum parallelism

Quantum computing which is not a reality today, when made possible will crack the DES using plaintext, ciphertext pair. Quantum algorithm makes use of the possibility to simultaneously manipulate coherent quantum superpositions of input states (Quantum parallelism). Assume we have a plaintext and the corresponding cipher text. We shall determine the DES key using the power of quantum computing. Quantum algorithm by Grover [6] to search a database can be used for this purpose.

Step 1: Load plaintext and ciphertext

Step 2: Load 256 combination of keys in quantum memory

Step 3: Calculate cipher text using each key

and compare it with the existing ciphertext
Step 4: For a particular quantum state a match is found. The corresponding key is the DES key

Because of quantum superposition, the computation takes place on all 256 state simultaneously. Once we have the secret key, any further communication between the two parties can be found out by deciphering the text with the known key.

Molecular computers

Due to advances in molecular biology it is nowadays possible to create a soup of roughly 10^{17} DNA strands that fits in a liter of water. Adleman [1,2] has shown that each DNA strand can be used to perform computations. Thus, a small test tube containing DNA strands seems to have more computing power than the most powerful parallel computers. The paper "Breaking DES Using a Molecular Computer" by Dan Boneh, Christopher Dunworth, Richard J. Lipton [6] estimates that given one arbitrary(plaintext, ciphertext) pair, one can recover the DES key in about 4 months of work. Further, they say that this method could be generalized to break any cryptosystem which uses keys of length less than 64 bits. What they mean by "breaking DES" is, given one plaintext, ciphertext pair, they can find a key mapping the plaintext to ciphertext.

The above said attacks to DES will fail in HDES. HDES maps a 56 bit plaintext to a 64 bit ciphertext using a 56 bit key. 8 random bits are added to the plaintext and placed in particular position to maximize diffusion. As a result a 56 bit plaintext for the same key will map to many 64 bit ciphertext. When executed for the next time, will again give different ciphertexts. Thus plaintext cipher text attack and differential attack fails in HDES

scheme.

The Security of HDES

The security of HDES relies on the random number. However, it seems very hard to find such a random number generator in the deterministic world of computers [7].

Random Number Generator

Here we describe a function which generates true random numbers. Random number generation depends on three important factors:

1. Random seed
2. Range of the seed
3. Speed of execution

Random seed

Usually computer timer is used for random number generation. When computer timer is used we have $60 \times 60 \times 24 = 86400$ seconds per day. i.e., 86400 values. If we use timer as random seed, at the same time the next day, we will get same numbers. Hence we call it pseudo random number generator. If we make use of date and time together as the seed, never in our lifetime, the same seed will repeat. Hence it is perfectly random and unpredictable. The function for randomized seed generation is given below:

```
FUNCTION SEED AS QUAD
DIM S,S1,S2 AS QUAD
DIM TEXT1 AS STRING
DIM N1,N2,N3 AS LONG
RANDOMIZE TIMER
TEXT1 = DATE$
N1 = VAL(MID$(TEXT1,1,2))
N2 = VAL(MID$(TEXT1,4,2))
N3 = VAL(MID$(TEXT1,7,4))
S1 = (CQUD(RND*N1)+1) *
(CQUD(RND*N2)+2) *
```

```
(CQUD(RND*N3)+3) +4 ))
S2 = (CQUD(RND*S1)+5) * S1
S = S2 + CQUD(RND*TIMER+1) *
CQUD(RND*N3+N2)
SEED = S
END FUNCTION
```

Range of the seed

If the range of the seed is larger, the seed is more randomized. If the seed is between 1 and 5, the seed is not considered to be at random. Here we talk about the degree of randomness, when the range is large, the probability of selecting one particular seed is less.

Speed of execution

Speed of execution is inversely proportional to the degree of randomness. I compared generating random numbers with my old PC XT 286 and my new Pentium IV PC. The degree of randomness with my old PC was comparatively high. So when we include a permissible delay while generating random numbers, we get random numbers with high degree of randomness.

Conclusion

We conclude that Homophonic DES is a better choice and provides greater resistance compared to DES. Random number generators have great impact on the security of this system and the suggested RNG generates a non deterministic output each time. Also we generalize that incorporating such RNG's to any homophonic systems will increase the security of the system.

References

- [1] L.Adleman, "Molecular computation of solutions to combinatorial problems", Science.v.266.Nov.1994.1021-24[2]
L.Adleman, P.Rothmund, S.Roweis,

- E. Winfree, "On applying molecular computation to the Data Encryption Standard", 2nd DIMACS workshop on DNA based computers, Princeton. 1996.28-48
- [3] E. Biham and A. Shamir, "Differential cryptosystems, Advances in cryptology – CRYPTO'90 Proceedings, Springer, Berlin, 1991, pp 2-21
- [4] E. Biham and A. Shamir, "Differential cryptanalysis of Snefru, Khafre, REDOC – II, LOKI, and Lucifer, Advances in Cryptology - CRYPTO'91 Proceedings, Springer, Berlin, 1992, pp. 156-171
- [5] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16 round DES, Advances in cryptology – CRYPTO'92 Proceedings, Springer, Berlin, 1993, pp487-96
- [6] D. Boneh, C. Dunworth, R. Lipton. "Breaking DES using a molecular computer", 1st DIMACS workshop on DNA based computers, Princeton, in DIMACS series, Vol.27(1996)37-65
- [7] D. Knuth, "The Art of computer programming", Vol.2, Semi numerical algorithms, 2nd ed., Addison Wesley, 1981.
- [8] National Bureau of Standards: "Data Encryption standard", US Department of Commerce, FIPS, pub. 46 January, 1977.
- [9] Tsu-Miin Hsieh, Yi-Shiung Yeh, Yung-Cheng Hsieh, Chan-Chi Wang, "A Homophonic DES", Information Processing Letters 66(1998) 317-320



Geetha Sivakumar is a Research Scholar under the guidance of Dr. M. Thiyagarajan, Professor, at SASTRA Deemed University School of Computing, Thirumalaisamudram, Thanjavur. He may be reached at gitaskumar@yahoo.com.



We speak CVE®!

The INFOSEC Evaluation Methodology (IEM) is NSA's hands-on process for conducting evaluations of customer networks utilizing common technical evaluation tools. Students can expect to learn an easily repeatable methodology that provides each customer a roadmap for addressing their security concerns and increasing their security posture.

7/31 - 8/1, 2006 **Las Vegas, NV**
Black Hat

9/28 - 9/29, 2006 **Atlanta, GA**

10/11 - 10/12, 2006 **Knoxville, TN**

10/26 - 10/27, 2006 **San Diego, CA**

11/8 - 11/9, 2006 **Las Vegas, NV**

To register for one of these courses or to get further information, please contact us at:

(719) 488-4500

info@securityhorizon.com

<http://www.securityhorizon.com>



The Future

Brought To You By Inspiration—and Coffee

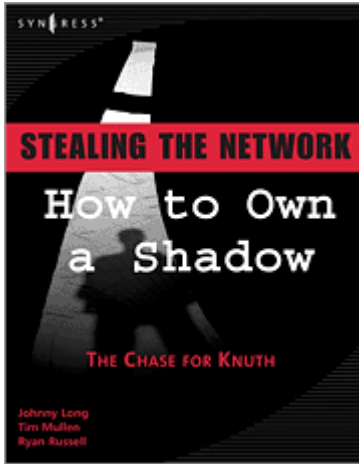
The University of Advancing Technology (UAT) provides students a diverse, exhilarating environment where the best elements of a college education collide with an unrivaled passion for advancing technology. It's an innovative educational model that gives talented individuals an opportunity to create the future with a complete immersion in the rapidly expanding technological universe. We need the very best professors/instructors to continue this vision. If you are committed to expanding the field of technology (online or on-campus) through research and sharing of knowledge—challenging, educating and empowering students, we invite you to explore the teaching opportunities at UAT.



Learn more.

www.uat.edu or email Dave Bolman,
Provost at dbolman@uat.edu

Book Excerpt: Syngress Publishing



The following is a book excerpt from *Stealing the Network: How to Own a Shadow* featuring Johnny Long, Tim Mullen, and Ryan Russell, ISBN: 1-59749-081-4 to be released by Syngress Publishing



This was easily the most emotionally real conversation Pawn had ever had with anyone, male or female, and Pawn was starting to itch from it. He was all too eager to get absorbed in the next challenge.

r: good deal. welcome to the next level!
r: this time, u r really gonna need a proxy
p: Sure, sure.
r: ok, we'll see brainy boy
r: http://yt.technet.edu
p: Got it.
r: u want some help?
p: Already?
r: lol

Pawn loaded the page. It looked familiar. Rafa certainly didn't waste any energy on graphics.



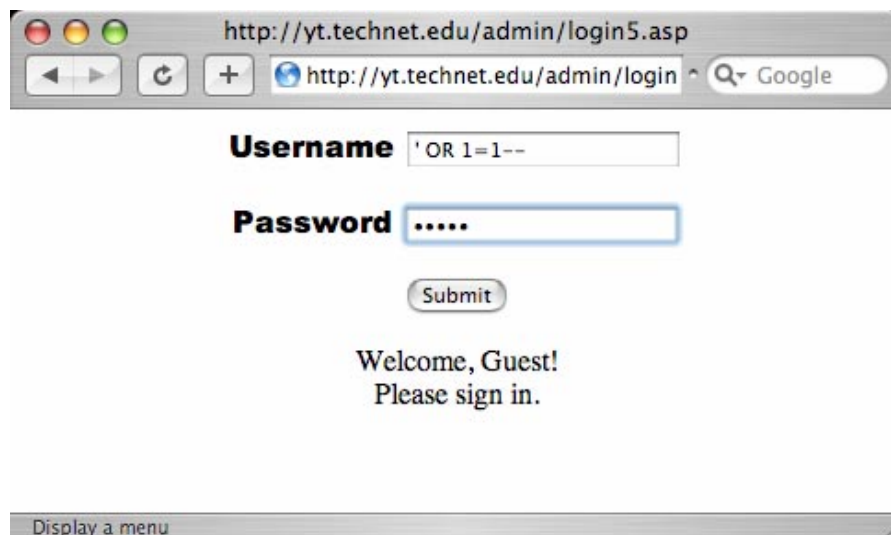
Book Excerpt: Syngress Publishing, cont.

Pawn viewed the page source. The body of the page was very simple.

```
<frameset rows="20%,80%" border="0">
<frame name = "header" src="header.html">
<frameset cols="20%,80%">
<frame name="login" src="admin/login5.asp">
<frame name="body" src="secret.asp">
</frameset>
</frameset>
```

The page loaded a header, the now-familiar login5.asp script, and a page called secret.asp into a set of frames. When loaded by itself, the secret.asp script prompted him to log in.

He cracked open Paros and loaded the *login5.asp* page. The results were exactly what he had seen with the POST exercise. Paros' screen updated, and now seemed to understand that the *admin/login5.asp* script used POST instead of GET. Pawn entered a single-quote into the username field and clicked *Submit*. The page complained that the username was too short. Pawn knew how this worked. He typed in his first injection, which now felt like habit, and entered a five-character password.

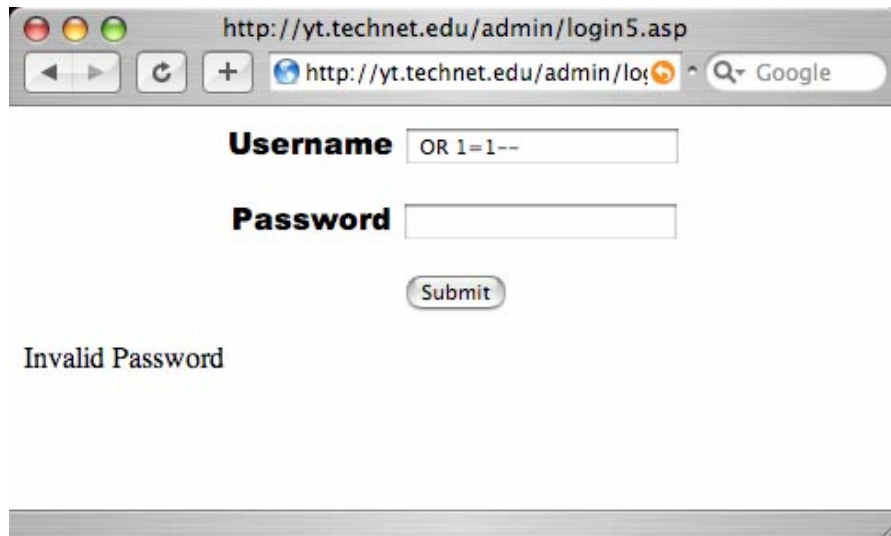


Pawn wasn't prepared for the result. Not only did the application not grant him access, as the previous applications had, it complained about an invalid password, and seemed to delete the single-quote from the username field!

Sure enough, the username field was missing the single quote! Pawn sat back in his chair and thought through the problem.

What can I use to get past this?

Book Excerpt: Syngress Publishing, cont.



He quickly sat up and started entering other characters, five at a time, into the username field. His choices began sanely enough as he tried characters other than single quotes that might error out the SQL.

```
.....  
'''''''  
-----  
,----
```

Eventually, Pawn's input looked more like obscene ASCII art.

```
@#$$@  
$^%$%^  
^&*$^%  
#$%#$$%
```

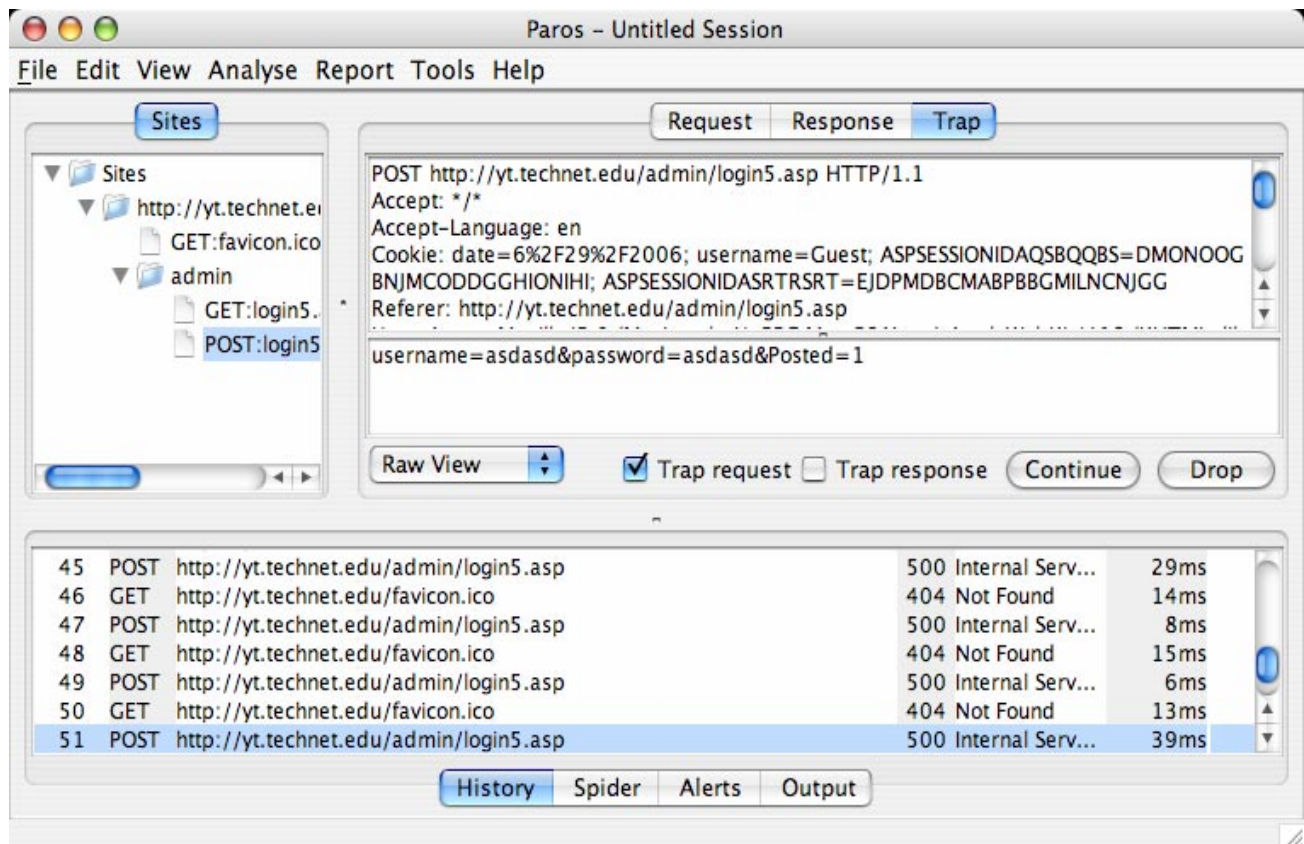
One after another, he pounded character after character into the username field, but it was no use. The username field was bulletproof. He continued the exercise again, this time focusing on the password field. Same deal. The password field seemed immune to any nasty characters. He sat back, clasped his hands behind his neck, and looked at the ceiling, trying to get a leash on his frustration.

One little puzzle after another, he reminded himself. He sat up and looked at his screen. The browser window seemed to glare at him, challenging him to jump back in. He was about to give in and start pounding the login field, when he saw the corner of the Paros window, sitting idle behind the browser.

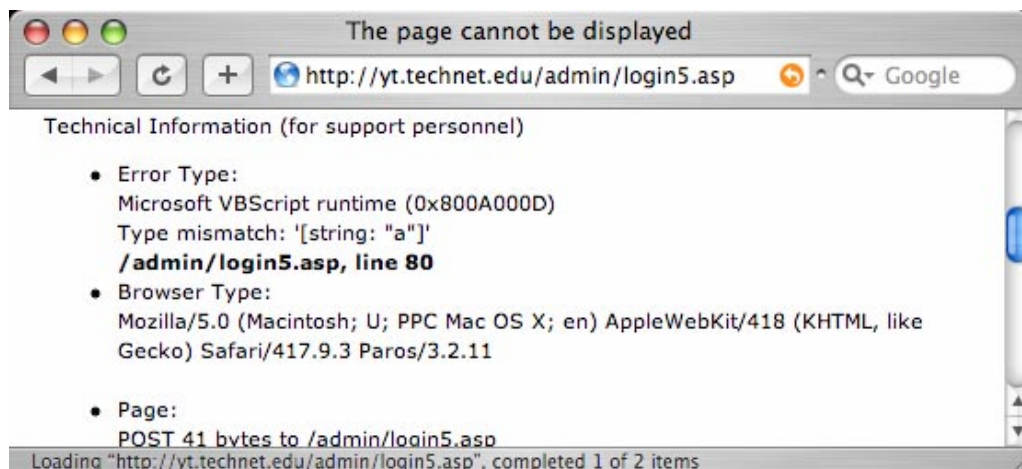
Paros.

He clicked through the history concentrating on the various POSTS he had sent to the login script.

Book Excerpt: Syngress Publishing, cont.



Having already abused the username and password fields, Pawn turned his attention to the field called *Posted*. He changed it to the letter 'a', and was greeted with his first error message from the application.



Pawn's brow furrowed as he read the message. It didn't look at all like the SQL errors he was used to seeing. This was a *VBScript* error. Pawn had caused some sort of problem with the `login5.asp` script itself. He Googled the error message itself, and learned that this error could be caused by comparing two different types of data. The script was obviously expecting a number (it had set the value to one when he posted the form) and Pawn had entered a

Book Excerpt: Syngress Publishing, cont.

character.

There has to be something here. This is the way in.

He pounded character after character into the *Posted* field, trying to get something other than a type mismatch error. After dropping 30 different hex values into the field, Pawn stopped. He realized he was flailing again.

There has to be a better way to test values here.

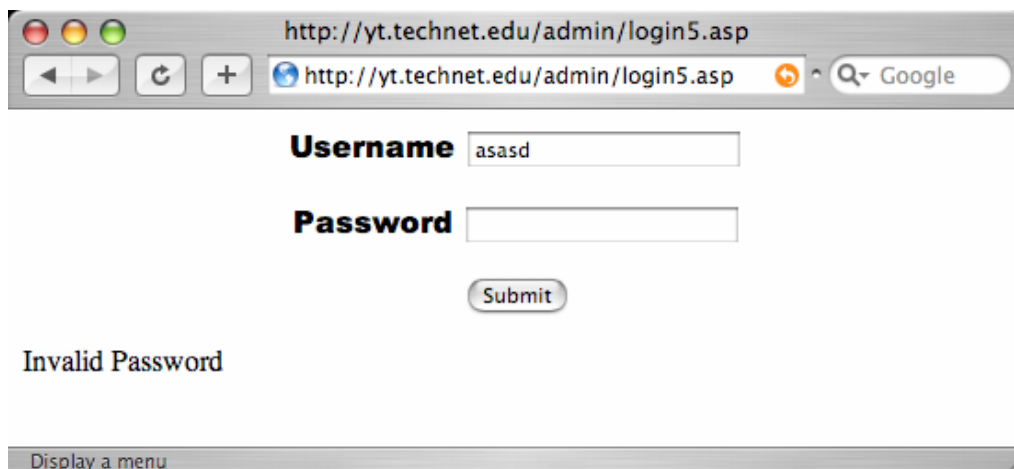
He considered looking for a tool that would help, but realized he might be on the wrong track. This wasn't an SQL injection point. This was something different. He looked at the request in Paros.

This has to be the way in. I've tried breaking every POST field and I've gotten nowhere...

Looking through the request, Pawn's gaze settled on the header fields. There, in his request, he saw a cookie. The server had sent him a cookie, and his browser was spitting back before every POST. Although cookies were a common occurrence, none of the other challenge servers threw him a cookie. He copied the cookie from Paros, and pasted them into a text editor.

```
Cookie: date=6%2F29%2F2006; username=Guest;  
ASPSESSIONIDAQSBQQBS=DMONOOGBNJMCODDGGHIONIHI;  
ASPSESSIONIDASRTRSRT=EJDPMDBCMABPBGMILNCNJGG
```

A quick Google search revealed that the ASP Session tokens were used to keep a user's state between sessions. These were wrapped in all sorts of crypto, and Pawn wasn't up for a battle against crypto, so he focused on *date* and *username*. *Date* was a straightforward thing, but *username* was curious. The value was set to *Guest* and Pawn remembered that this was the username that was in the form when he first loaded the login page. He changed the value to *test* and continued his session.



Book Excerpt: Syngress Publishing, cont.

The page didn't show a single trace of *test* value. Pawn threw a single quote in as the cookie's *username*. Same thing. Invalid Password.

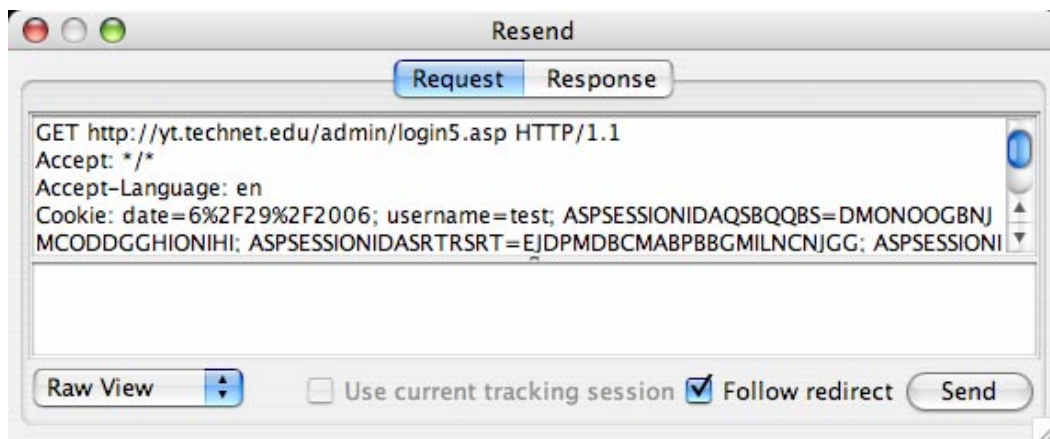
Why isn't the cookie's value getting used anymore?

Pawn flushed his browser's cache, and loaded up the login page again. There it was, plain as day. "Welcome, Guest! Please sign in." Paros saw the initial load as a *GET* request. No data had been sent via POST, and only a cookie had been sent.

The application uses the cookie to populate the username field when the user first visits the page.

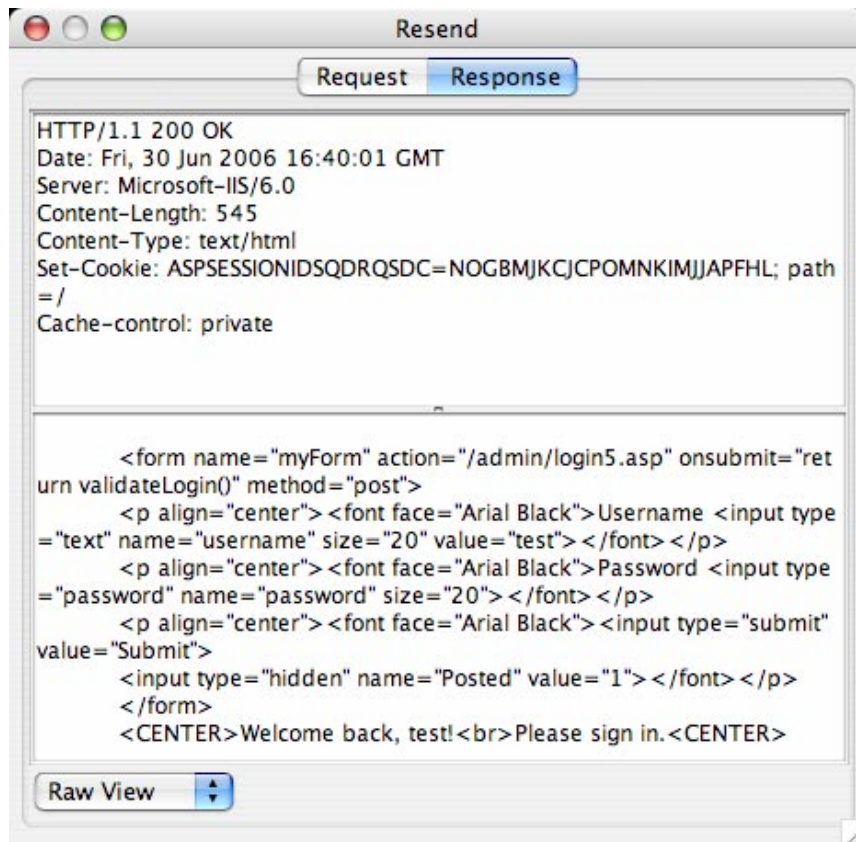
This trick was used on websites all over the Internet. It provided a (silly) way to make a user feel welcome, but only if it reflected their actual username. After a successful login, the cookie would most likely be updated to reflect their real username.

Pawn instinctively right clicked on the last GET request in Paros, and saw a menu he didn't even know was there. At the top of the menu was the *Resend...* option. Pawn smiled. "Resend?"



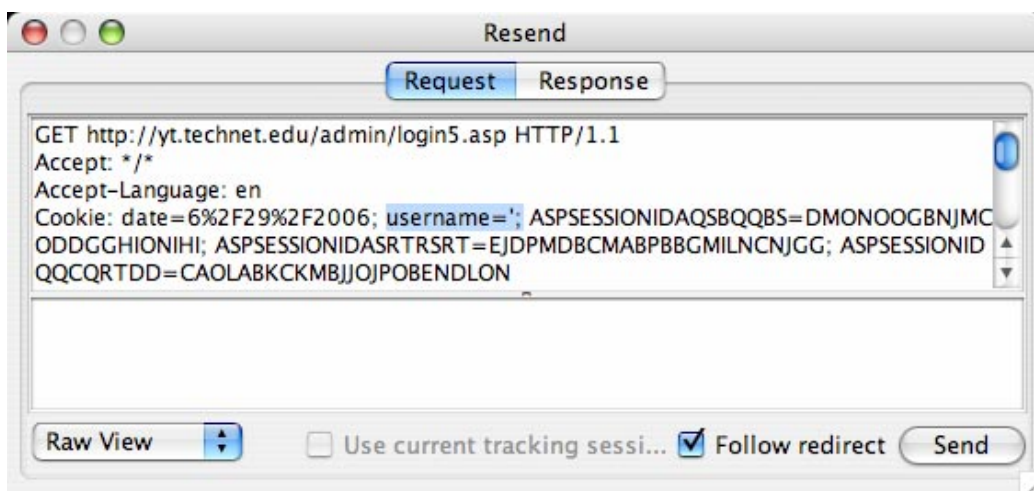
Pawn changed *Guest* to *test* and clicked *Send*. The request was sent to the server. Pawn felt silly that he hadn't found this option earlier. It would have saved him a ton of time. The page flipped to show the response, and Pawn could tell from the HTML that server took his modified cookie.

Book Excerpt: Syngress Publishing, cont.



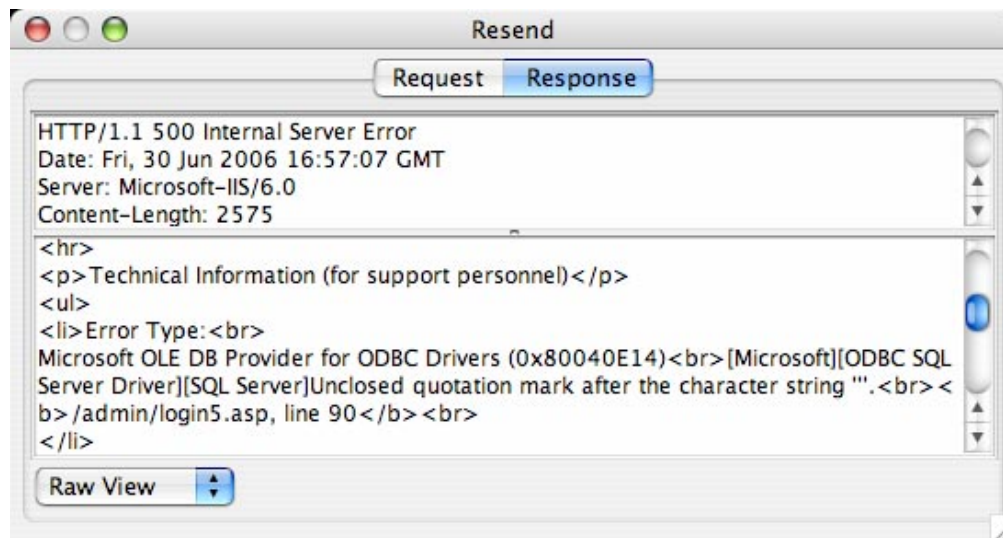
"Welcome back, test! Please sign in."

Pawn smiled, and instinctively switched into abuse mode. "Now, let's see if I can break you." He set the cookie's *username* value to a single quote, and resent the request.



Ugly HTML that it was, the response was nothing short of beautiful.

Book Excerpt: Syngress Publishing, cont.



Pawn's friend, the *500 Internal Server Error* was back in town, and Pawn was back in business. Forgotten were all the little bumps along the way, and the thrashing that had consumed so much time. All Pawn could focus on was the beauty of this error, and how its delivery had been the result of solving one little puzzle after another. Although he hadn't finished the challenge, it would all certainly be a breeze after this.

There was no celebration. Pawn didn't jump up out of his chair and dance around like a madman, or even lean back and throw his arms in the air. There wasn't even so much as a victorious chair spinning to mark the occasion.

There was just this look. Pawn's chin rested on his chest, his brow was furrowed, and his eyes rolled upwards, straining so far that his eyebrows painted a dark horizon line low across his line of vision. He did not smile. A faint smirk was the only indication that he was not furiously angry.

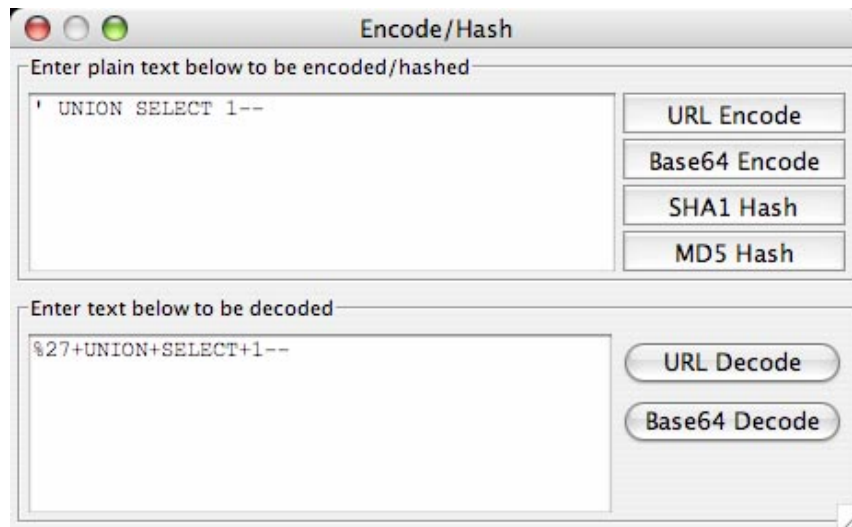
Pawn clicked the *Tools* menu in Paros, and found the *Encoder/Hash* tool. He didn't even pause to realize he had never used it before. Somewhere deep in his mind, he remembered seeing it there when he was flailing, and he knew its purpose. He knew almost without knowing.

Pawn began typing, hard, into the encoder. He loaded up the encoder with the first injection. He was typing fast, but not flailing. Not now. He knew what he had to do and he was running on instinct.

Recon. How many fields returned from the original select?

He hammered the injection into the encoder and clicked *URL Encode*.

Book Excerpt: Syngress Publishing, cont.



He paused just long enough to know he hated how the encoder had used plus signs instead of the hex-encoded %20. He knew the plus signs could cause problems if he needed an actual plus sign in his SQL, but he didn't. He let the tool's encoded text ride.

He copied the encoded text and slammed it into *username*. *Cookie updated*. He clicked *Send*. He knew what the response would be before it even came.

All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of expressions in their target lists.

Pawn didn't even flinch at the error. *Balance it*.

He updated the injection. *More fields*.

```
' UNION SELECT 1--
```

became

```
' UNION SELECT 1,1--
```

He clicked *Send*. Still imbalanced.

```
' UNION SELECT 1,1,1--
```

Pawn wasn't phased by the repetition. It was expected. He knew that eventually it would hit.

Send.

```
' UNION SELECT 1,1,1,1--
```

Book Excerpt: Syngress Publishing, cont.

And then it did. Nestled deep in the HTML response, the server greeted him. "Welcome back, 1!"

Four fields in the original select. What are their names?

Pawn was typing so fast and so fluidly, that it seemed as if he had done this a hundred times before. He manually typed the HAVING clause into the request, and eyeballed it.

```
%27+HAVING%201=1--
```

Wrong. The equal sign is used to separate names from values in the cookie. Encode it.

```
%27+HAVING%201%3D1--
```

The cookie updated, and the error avoided before it could even occur, Pawn clicked *Send*. Another beautiful response.

```
Column 'cookies.username' is invalid in the select list because
it is not contained in either an aggregate function or the GROUP
BY clause.
```

The table name is cookies and the first column name is username. Throw it into the injection and find out what's next.

He updated the injection by hand.

```
%27+group+by+cookies.username+having%201%3D1--
```

Send.

```
Column 'cookies.date' is invalid in the select list...
```

Good. cookies.date. Throw it in, find out what's next.

```
Column 'cookies.ip_address' is invalid in the select list...
```

Pawn continued ripping through the fields one after another until he had mapped out all the fields in the original SELECT statement. It didn't occur to him just how fast he was flying through this exercise. He had absorbed so much in the past two days, and it was beginning to show. He moved effortlessly into the extraction phase.

Let's get to that data.

Knowing he would use the function he had created in the last exercise, he threw together a test function that returned the number one. He knew that it was a good idea to get a test function in

Book Excerpt: Syngress Publishing, cont.

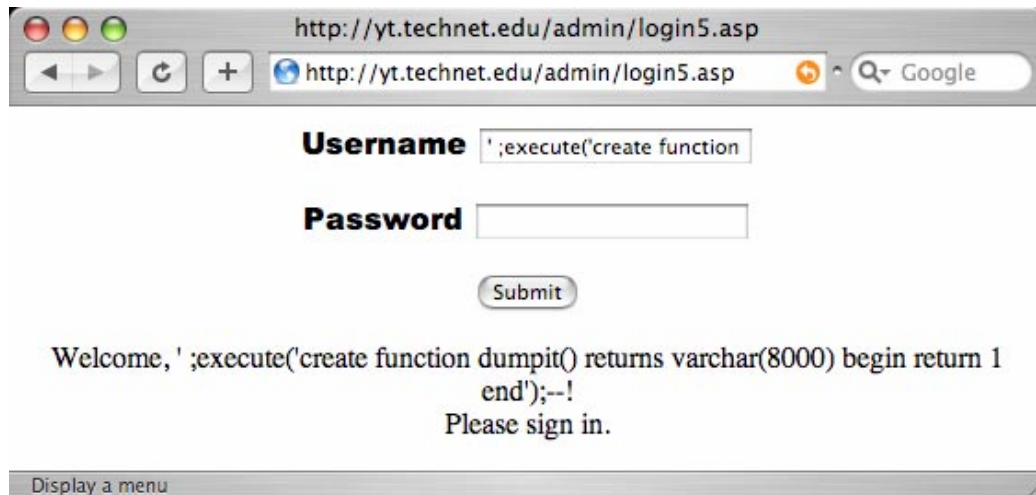
there before dumping the real one.

```
execute('create function dumpit() returns varchar(8000) begin
return 1 end');--
```

Wrapped in an execute statement since *CREATE* was supposed to be the first statement in a T-SQL batch, he URL encoded it and slapped it into the cookie.

```
username=%27%3Bexecute(create+function+dumpit%28%29+returns+varchar%288000%29+begin+return+1+end)%3B-;
```

He sent the request, and Pawn went full stop. He had to load the request in a browser to make sure he wasn't seeing things.



Rafa had been impressed with Pawn's progress so far; she had been more than impressed. Pawn's progress over the past few days had been nothing short of amazing, but Pawn didn't see it that way.

If Pawn didn't *excel* at the challenges, he felt defeated. Creating a custom function in the last challenge was seriously sweet. It dumped all the records in one shot, exposing the database's contents through a single URL. Pawn could have simply dumped the record one at a time, but he refused to think in those terms.

That's exactly what an n00b would do.

Pawn's test function wouldn't take, no matter what he did. He even tried jamming in the real function, the one that would dump all the records from the cookie table. The result was the same. The page stupidly echoed what he injected.

If Pawn had stepped back and thought about it, he would have realized that the server program

Book Excerpt: Syngress Publishing, cont.



was simply reacting to the SQL it was handed. He would have realized that just like in the first challenge, a page isn't always what it appears, and he would have used these subtleties to diagnose his problem. Dead set on determining what was wrong, Pawn would spend the next two hours trying permutation after permutation, over and over again, trying to get the server to execute his SQL, but time after time, it would just echo it. Pawn entered that dark place in his mind that suggested that the only logical solution was that something had broken on the server, and that his mind and his theories were still completely logical.

After hours of seeing the server echo back his SQL, Pawn never thought to break his SQL to see if syntax errors were still reported. Had he seen this, he would have realized that the "echoing" feature of the application was trying to tell him something. And then, he would have realized that Rafa had disallowed CREATE, and that a silent, subtle echoing "Welcome" was all he would see of the error.

Pawn's anger built into a fury that led him to the heavy bag. Although his years of Ninjutsu shone through his impeccable form, after twenty intense minutes, his rage left him in a soaking wet, utterly exhausted heap in the middle of his dojo floor.

As he closed his eyes in what would be a haunted night's sleep, he could make out the faintest voice of reason.

Your anger's going to be the death of you.

Look for the fourth and final book in the Stealing series in stores August 2006.



Syngress Publishing (www.syngress.com), headquartered in Rockland, Massachusetts, is an independent publisher of print and electronic reference materials for Information Technology professionals seeking skill enhancement and career advancement. Distributed throughout Europe, Asia, and the U.S. and Canada, Syngress titles have been translated into twenty languages. For more information on Syngress products, contact Amy Pedersen at 781-681-5151 or email amy@syngress.com. Syngress books are distributed in the United States and Canada by O'Reilly Media, Inc.



Examine, expose, and exploit your vulnerabilities before an attacker does.



Introducing the first integrated vulnerability assessment and penetration testing tool.

SAINTexploit™, developed by the maker of the award-winning SAINT scanner, gives you the ultimate resource to demonstrate the security—or vulnerability—of your network before an attacker gets there first.

SAINTexploit™ integrates seamlessly to:

- Examine potentially vulnerable services
- Expose points where an attacker could breach the network
- Exploit the vulnerability to prove its existence without a doubt

Don't just expose your vulnerabilities, exploit them.

To learn more, visit www.saintcorporation.com/saint_exploit.html or contact Billy Austin at 1-800-596-2006 x0119 or austin@saintcorporation.com.

There has been a lot of press on the security, and sometimes the severe lack thereof, of 802.11x Wifi networks...



The popularity of the technology is both overwhelming and understandable. Vendors have finally hit that sweet spot between effective technology and its ease of use. But with all the news, books and articles written on the security concerns of wireless networks, very few actually touch upon the wireless video security that uses the same technology. This article is intended to be an introduction to the concepts and potential security concerns of a wireless video system. For more detailed information, I recommend that you do some research beyond this article and also look for the new book from Syngress Press, *WarDriving & Wireless Penetration Testing* (ISBN: 159749111X), coming out in October of 2006.

The Basics

Wireless cameras are becoming more and more popular. Dozens of vendors with names like Sony, D-Link, Toshiba, and Cisco have all jumped into the market with their various offerings. The key to the technology is ease of use. Users can buy the product, install it within just a few minutes, and be completely operational. The first offerings came out as Baby Cams, allowing parents to keep a wary eye on their sleeping children. Soon, the technology took the leap into the business world, allowing organizations to add video security to their organization without running cables through all the walls, as is required with Closed Circuit systems (CCTV).



Figure 1: Wireless Baby Cam

Once these cameras are installed, their transmissions are public knowledge. Any individual with the right hardware/software can drive by your house and determine whether you're running a wireless network. Wireless devices tend to be fairly chatty, and as such, will transmit information about what wireless hardware you're using, the name of your network, the internal network address space, and more. A motivated attacker can drive through a neighborhood and pick his/her targets based on this information.



Figure 2: Wireless Infrared Camera

Wireless cameras are no different than wireless networks. They still send signals indicating their existence. In most cases, the signals are not encrypted at all, allowing someone with the right hardware to actually see and hear what the camera does. In terms of security, this could be a huge concern. If you're a large corporate entity using these devices, you could be opening yourself up to corporate espionage or break-in. For home users, you're transmitting pictures of your home, its contents, and your children to the area surrounding your home. Depending on the conditions, these signals could be picked

up from up to two miles away.

The Details

The 802.11b standard actually defines 14 channels within the 2.4 GHz range that can be used, but the FCC approved frequency range for 802.11 wireless networking within the United States is broken into 11 channels. Because the range is for use in products, a user can find any number of devices that are set to communicate at these frequencies. This includes cordless phones, wireless access points, and wireless computer mice. The actual frequencies for each channel are listed in the Table 1.

Channel 1	2.412 GHz		Channel 7	2.442 GHz
Channel 2	2.417 GHz		Channel 8	2.447 GHz
Channel 3	2.422 GHz		Channel 9	2.452 GHz
Channel 4	2.427 GHz		Channel 10	2.457 GHz
Channel 5	2.432 GHz		Channel 11	2.462 GHz
Channel 6	2.437 GHz			

Table 1

The frequencies listed above are actually the center frequency in a slightly larger frequency range that each channel uses. For example, Channel 1 operates at 2.412 GHz, but there is a 5Hz frequency buffer on each side of that helping to avoid interference between the channels. If you notice, Channel 2 operates at 2.417, which is exactly 5Hz above Channel 1. But this also means that there is an overlap in the buffer zone between the two channels. Due to this, your wireless networks, 2.4GHz phones systems, or other wifi enabled devices may interfere with your wireless security camera system. If you have these issues, consider modifying the device to operate on a different channel.



Figure 3: X10's Xcam2

User Hardware

Newer wireless video hardware has a more professional look and feel. They provide functionality as security cameras, baby monitors, or webcams. As an added perk, many of them now come with built-in web servers, dynamic DNS support, and remote control capabilities. They can be built to look like normal video cameras, clock radios, speakers, or even teddy bears.



Figure 4: Clock Radio Camera



Figure 5: Teddy Bear Cam

In the simplest examples of this hardware, all you have to do is plug in the device and put batteries into the remote viewing unit. In more complex setups, like the ones from X10, you may need to plug a receiver into a VCR or other recording device. But in almost all cases, the hardware is fairly inexpensive to purchase and easy to operate. Don't let the

price fool you, however, the signals from these devices (yes, even the Teddy Bear Cam) can reach up to 2 miles from their source in a clear line-of-sight situation.

Snooping Hardware

Wireless security cameras haven't come under the same scrutiny that wireless networks have undergone in the last several years. Because of this, the technology is relatively insecure and vulnerable to snooping. In the most extreme cases, it's technologically possible to introduce your own recorded signal to the user's recording device. In these rare cases, the video signal being recorded by the recording or monitoring device may not be the same image actually being seen through the camera's eyes.

As an example, consider a home user that is on vacation but monitors their home through a website fitted with their wireless security camera. It's possible to send a stronger (think louder) signal to the receiver attached to the computer, showing images of the home in a safe and secure condition; all the while thieves have already broken into the home.

One of the simplest means for snooping on these signals comes from ICOM. If you're a Ham Radio buff, then you've already heard of ICOM. If you're not, you should know that ICOM is in the business of building transceivers for use on public bands. This includes some frequency scanning hardware, such as the ICOM IC-R3. The IC-R3 is an upgrade to an older model scanner, the IC-R2. But when they released the IC-R3, ICOM added one huge benefit to the device that no other vendor had yet added; a small color TFT screen, allowing users to view video signals in public bands. It runs about \$300 USD and has a telescoping antenna (BNC connector) that can easily be replaced with a more

powerful directional antenna. This is probably the most convenient means for eavesdropping on someone else's video transmissions.



Figure 6: ICOM IC-R3



Figure 7: X10 Video Receiver

The X10 setup is slightly more complex, but the video transmissions can still be picked up via the IC-R3, in most cases. In some cases, the snoop might opt to buy the actual video receiver that X10 recommends for their product. The signal from the cameras is picked up by the receiver and then sent across an RCA cable to the appropriate output, such as a VCR or television monitor. Because of the added number of components, such as a power inverter for use in a moving vehicle, the X10 setup ends costing about the same as the IC-R3.

Protection Mechanisms

Unfortunately, up to this point in time, not a lot of progress has been made on protecting consumer grade wireless video technology. The demand for wireless networking has driven much of the research and investment from a security perspective, leaving wireless video out in the cold. Many of the projects being developed and implemented today

simply won't work for home users and smaller organizations. X10 products include secure encryption algorithms within their remote control software and web applications. And although this protects the digital data as it traverses the Internet, it doesn't protect the actual wireless signal that is traveling through the air.

Wireless cameras should be used with a measured dose of paranoia. We're providing a short list of things to take into consideration. But until vendors offer end users more secure possibilities for wireless video, you should always be careful when utilizing this technology.



- o Don't leave the cameras pointed at the kids if there are no adults in the home.
- o Don't use the cameras in situations that you wouldn't want a perfect stranger to see.
- o Try to avoid pointing cameras at important assets or valuables in your home.
- o When you initially set up your wireless video system, scan your local area to find out what channels the other neighbors may already be utilizing for wireless devices, and pick something different.
- o Disable the Internet capabilities, web server, or remote control aspects of the cameras if you're at home and not using them.
- o If your camera comes with the ability to turn down the transmission strength of the cameras, adjust the signal power to the point that the receiver just loses the image, then tune slightly up again until the receiver regains clear signal.
- o Finally, don't rely solely on these products. Nothing beats getting up off the couch and checking on the kids in person.



Russ Rogers is the CEO and CTO of Security Horizon, Inc. He is the Editor in Chief of the Security Journal and may be reached at russ@securityhorizon.com

Discussion of the key elements of Electronic Signatures enabled by Public Key Infrastructure within the Title 21 C.F.R. Part 11 context



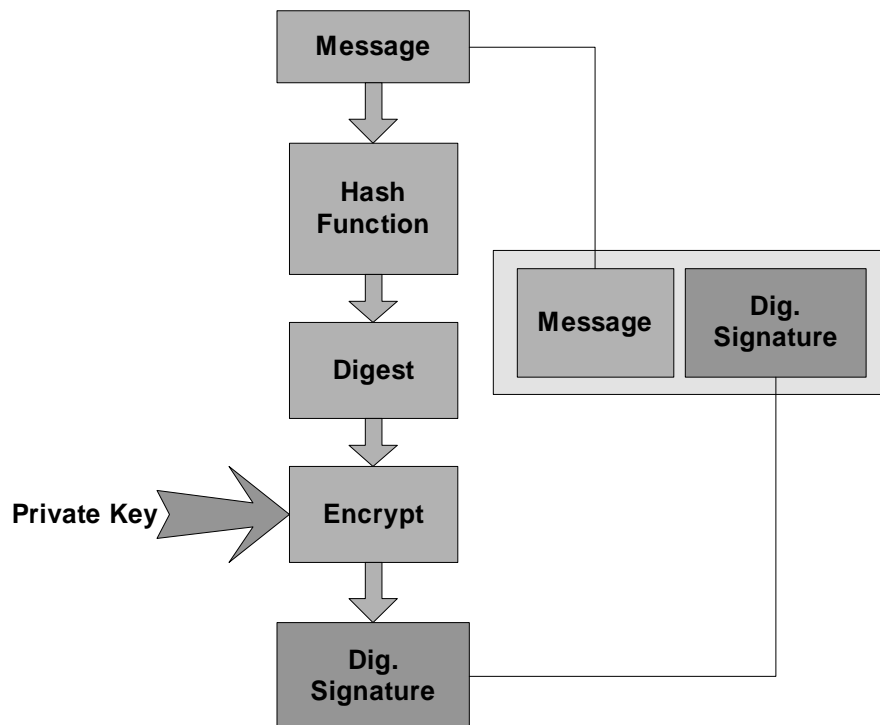
The following is Part 2 of a two-part article.

Understanding the hashing function and digital signature

A digital signature is used to verify the identity of the sender and to ensure that the contents of the message have not been tampered with during transit from sender to receiver. The message digital signature ensures that the body of the message (text) has not been altered after it was sent.

Messages are digitally signed with a secure “hash”. A “hash” is a numerical output of a cipher or encryption algorithm⁷. The Secure Hash Algorithm, version 1 (SHA-1) or the Message Digest, version 5 (MD5) protocols are standards to create such products.

As illustrated below, the sender uses a private key to create the message digest. The sender’s private key corresponds to a public key (private/public key pair) used by the receiver to validate the authenticity of the message.



The hash function is used to produce a one-way fixed size hash value of the message. This one-way strong, or message digest, is a checksum of the message. The message digest is then encrypted using the sender's private key.

As message authenticity is provided by the use of the sender's private key, that key must be securely stored. Loss or compromise of the private key can result in the digital signing of fraudulent and erroneous messages. As the private key is a simple BASE64 file, it must be stored and safeguarded properly (in a very secure manner). Some commercial implementations of smart-cards (chip on the card) require a password or personal identification number (PIN) to access the card and the private key.

Summary of PKI complexity and organization response

When an organization or institution deploys a PKI it is to support authorized users needs for data confidentiality and integrity. Many times the PKI is embedded within the organization's existing infrastructure. For example, Microsoft Windows 2000 server operating system has an embedded PKI within the operating system to provide user authentication.

Certificate management can be cumbersome and complex. This is a drawback of PKI. Those organizations that decide to commit to the deployment of a PKI must provide the adequate operational, administrative and management (OAM) resources to operate the PKI. Many organizations do not have this type of OAM resources to make available to a PKI deployment.

Care should be taken by the management of an organization to understand the significance

of OAM resources to sustain a PKI.

Digital signatures and 21 CFR 11

PKI-enabled digital signatures provide authentication and non-repudiation of those documents that have been "signed" with a user's private key credential. The legality of such a signature has been upheld as "legally binding". Thus, for the purposes of submitting voluminous regulatory, compliance and oversight documents to government agencies, digital signatures provide the mechanics.

Each year companies, institutions and research organizations must submit hundreds, if not thousands, of documents to the FDA. Presently, the majority of life sciences companies must rely on the submission of paper-based documents. The wastefulness of paper-based document submission (in terms of timeliness, production, transportation, etc.) is obvious. The return on investment (ROI) of electronic document submission techniques vs. paper-based submissions may be impossible to calculate.

The recent "SAFE-BioPharma" initiative has provided renewed interest in the feasibility of electronic document submission that is legally binding. "SAFE-BioPharma" promises to remove many of the protocol, interface and governance issues that defy interoperability between different organizations.



David Sweigert, CISSP, PMP, is the Managing Director of the Security Alliance, LLC. He is a U.S. Air Force veteran and has provided PKI and digital signature support to a number of diverse organizations. He can be reached at dgsweigert@juno.com



“Hello, my name is Brian and I am an information junkie”



This is hopefully the first installment of a regular feature for The Security Journal. I'm the geek in the office that walks around saying, "Have you seen this tool? It will really help with what you are doing..." There will be a wide range of tools discussed. Some will be just daily useful tools while others are specific to the INFOSEC community. If you have a really cool tool that you love, let me know.

I just attended graduation for my master's degree (only 13 years after getting my bachelor's). As usual there was the nap time interruption by a motivational speaker, I have no clue what her name was. Of the 10 minute speech there was one theme I remember, information overload.

Take a step back and watch the humans around you. How many of them have a PDA that keeps them constantly connected? Are they really being productive? How much information can one really handle? Most professionals have the problem of not being able to see the forest through the trees. We all have the problem of information overload.

The INFOSEC world is an ever changing place. As an INFOSEC professional I always worry about not knowing what the latest vulnerability is. My nightmare is having the response of "Huh?" when The Exec walks into my office asking, "How will the 0day exploit for ThingyMaBob affect us, and have you

protected our environment yet?"

On a standard work day, which is Sunday through Saturday, I keep track of about 50 websites and another 20 mailing lists. I could spend all day just reading this information. Being a junkie that isn't a real problem, unless I am expected to get work and honey do's completed. RSS has come to my rescue.

There are multiple definitions for what RSS stands for. I like Really Simple Syndication. For a long story about RSS and the different version and names visit Wikipedia¹. Using RSS gives me the ability to easily see if there were updates to a site and/or a specific piece.

The number of sites that offer RSS feeds is growing on a daily basis. If a site does not have an official RSS feed you can usually find an unofficial feed from a third party. With all of the different feeds the issue becomes how to view and manage the different feeds.

I had to find an RSS Aggregation tool². There are many tools available, along with plugins for popular browsing and email applications. When choosing a tool make sure it fits with your computing habits. Do you work only on one machine? Or is all your work done from your blackberry? Do you have the requirement for cross platform support?

I was looking for a tool that enabled me to see

Brian's Tool Corner, cont.

my feeds from my desktop, laptop, and PDA. Nice to have included the ability to view my feeds from other machines, sharing my feeds and exporting my feeds if I wanted to change applications. My RSS aggregation tool of choice is BlogLines³. This site fit all my requirements including the nice to have. The bonus cool tool was the ability to have a bookmark labeled "Sub with Bloglines". If I am on a site I want to add to my feeds list I just click this bookmark and it gets added.

Bloglines has significantly reduced the amount of time it takes for me to stay current. I have converted a number of friends to using Bloglines and so far all of them love it. I have created an example site for you to view.

<http://www.bloglines.com/public/SecurityJournal>

The feeds listed here are a good sample of what I read daily. If you have suggestions for other must read feeds send me an email.

References

- 1: http://en.wikipedia.org/wiki/RSS_%28file_format%29
- 2: http://en.wikipedia.org/wiki/News_aggregator
- 3: <http://www.bloglines.com>



Brian Kirouac is a Senior Security Consultant for Security Horizon, Inc. He may be reached at bkirouac@securityhorizon.com.



Your global information security experts

Advertisers!
Reach *THOUSANDS*
of readers every
quarter at an
outrageously low
price!!

The **Security Journal** is never offline, making your ad constantly available--ALL of our issues are still downloaded quarterly



Contact us at:
editor@securityhorizon.com
or
719-488-4500