Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Randomness and Entropy - An Introduction

This paper will attempt to bring together information pertaining to concepts and definitions of randomness and entropy. Through definition and example both the implications and applications within the Information Security industry will be shown, bringing a complex topic to light in a concise and understandable form.

Copyright SANS Institute
Author Retains Full Rights

AD

Beating the IPS?

STONESOFT
Can't be Beat

Learn
More

Randomness and Entropy – An Introduction

GSEC Practical v 1.4a – Option 1

Chris Thorn

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Table of Contents

1.0 Abstract	iii
2.0 Introduction	iii
2.1 Randomness in Internet communications demonstration	iii
2.3 Cryptosystems demonstration involving randomness	iv
3.0 Defining Randomness	v
3.1 Randomness or not randomness	v
3.2 Entropy History	vi
3.3 Entropy Defined	vii
4.0 Applying Entropy to Randomness	vii
4.1 Creating better random data	viii
5.0 Future considerations and Conclusions	ix
5.1 Looking Forward	x
5.2 Additional Resources	xi
6.0 Bibliography	xii

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

1.0 Abstract

This paper will attempt to bring together information pertaining to concepts and definitions of randomness and entropy. Through definition and example both the implications and applications within the Information Security industry will be shown, bringing a complex topic to light in a concise and understandable form.

2.0 Introduction

Within security today we see that encryption methodologies and cryptographic techniques have exploded in both complexity and usage. The application of these technologies is to keep secrets safe and secure, but there are pitfalls involved with utilizing cryptography. The question is: are these pitfalls, superficial or vital issues? It is common knowledge that some cryptography is implemented far better than that of others. This paper focuses on one of the vital components used in various security related technologies. This component is by nature complex and easily misunderstood. One may say that randomness plays a "key" part in most cryptosystems today, however, generally speaking it is very difficult to appreciate how many systems and technologies rely on the randomness of data.

The scope of this paper brings data from multiple sources and attempts to explain in simple terms the role that randomness plays within applications utilizing Information Security. Through the use of simple examples, definitions, theories, and future developments it is hoped to heighten awareness of the complexities and contributions of randomness.

In order to understand the part which randomness plays in Information Security, a few examples will be useful, one demonstrating randomness in internet communications and another revealing its role in a known cryptosystem.

2.1 Randomness in internet communications demonstration

When port scanning, system security analysts are shown the probabilities of sequence number prediction. The following is an example of sequence numbers, what they are, and why randomness is important.

First, here is a little brush-up on TCP/IP and how the whole communications occur from node to node. It is called aptly the three-way handshake; this is where one node contacts the other in a number of synchronization and acknowledgments leading to a dialogue. In order for the communication to continue, each packet is given a number so that the receiving end knows which packets belong to the dialogue that has been initiated. These are the sequence numbers. It is important that the sequence numbers for a given dialogue be unique so that the dialogue between the initial nodes in the communication remains exclusive.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

It wouldn't be too much of a stretch to visualize the results of someone figuring out the sequence numbers, especially if they were to assume the identity of one of the nodes and send data with the expected sequence number. This would be referred to as a man in the middle attack.

Now with that explanation out of the way the question becomes, what keeps the sequence numbers from being guessed by the man in the middle? The answer is randomness. Sequence numbers are formulated on a randomly generated number which is agreed upon within the three-way hand shake at the very beginning of a dialogue, after that, the sequence number is incremented by an amount determined by the quantity of data contained within. Here we see the problem, if a pattern can be discerned and the original random number can be guessed, the transmission is no longer protected

2.3 Cryptosystems demonstration involving randomness

Most modern encryptions rely on randomly generated keys; Netscape's Secure Socket Layer (SSL) was originally an example of this, SSL utilized a very large random number. Unfortunately, back in 1995 two researchers determined that the original keys were based on the time of day and the process id's. Once this was determined, using this information they were quickly able to determine and anticipate the keys thus rendering the SSL stream ineffective. This of course was not a trivial event, as banking institutions along with many online merchants relied heavily on SSL encryption to do business. Netscape's flaw was determined to be directly associated with the randomness of the key value, or the apparent lack thereof.[1]

In conclusion; these two simple examples show clearly that in both cases randomness stood at the heart of some very influential and highly utilized protocols. The making or breaking of the random data used in both situations is paramount for the protocols to be secure and useful.

Random numbers are used in many places within the realm of information technologies and also external industries such as the sciences. We find randomness at the heart of cryptosystems, games, simulations, scientific analysis and communications. Although random numbers are based on complex mathematics there are definitions which are simple to understand, yet many when searching, find few complete explanations.

[1] *Random Numbers – Russel Kay*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3.0 Defining Randomness

Now that we see the major role that randomness plays we need to look at randomness in more detail. Definitions can sometimes simplify our outlook on a broad topic, while increasing our understanding. Unfortunately, because of the complex mathematics involved, text book definitions for randomness can be somewhat overwhelming and difficult to summarize. Please note that the following details are stated in a simplified manor because detailing statistical data and associated formulas, is beyond the scope of this paper.

The term "Randomness" has a number of different definitions, the dictionary defines randomness as "An inexplicable misfeature; gratuitous inelegance"[2] and pertaining to mathematical statistics states "Of or relating to a type of circumstance or event that is described by a probability distribution"[3]. Knowing that our application of randomness comes from mathematical statistics we look at the mathematical definition most commonly shown by the Kolmogorov-Chaitin complexity which defines a random string as one, which has no shorter description than the string itself.[4] If a large string exists and cannot be compressed it is said to be random. Knowing this, any sequence of numbers created through the sole use of mathematical algorithms cannot be truly random.

3.1 Randomness or not randomness

This is the next issue, how do we tell when something is random? By the definition above we ascertain that random data cannot be compressed, so one should be able to conclude that attempting compression must be one crude way of determining if data is random. However there must exist levels of randomness leading towards true random data, some random data has been proved through use of mathematics and statistics to assume properties of being random, however, at some point show a discernable pattern. These levels in mechanical statistics are called entropy and some have gone so far as to create named complexity levels demonstrating randomness. Expressed below is a brief synopsis of each named complexity:

<u>Quantum Randomness</u>
Quantum Randomness is based on quantum events such as splitting atoms, deriving randomness through molecule movement, division of cells and such. This is expressed as the most random occurrences known to date, which means nobody has been able to discern any patterns so far.

---

[2] *The Free On-line Dictionary of Computing - Denis Howe*
[3] *The American Heritage® Dictionary of the English Language, Fourth Edition -Houghton Mifflin Company.*
[4] *Generating a truly Random Number – Leif Svalgaard*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Secret Entropy
We see examples of utilizing secret entropy in most random number generators, using input such as background noise, variations of electronic noise, or client input such as mouse or keyboard usage.

Obscure Complexity
The idea of obscurity is based on the appearance of complexity and misunderstanding of methodologies. An example of this would be using the time or process time frequency being reported by Microsoft Windows or other such monitoring tools.

Rand() Function
As it sounds this is a mathematical algorithm which spits out random data within tolerances and abilities such as mathematically generated randomness can be claimed.

Gee - Wow Complexity
Who names these things anyway? This is seemingly complex numbers that really are not that entropic, closely related to the obscurity complexity though completely unsubstantiated in quality of randomness. Chaos theory, Mandelbrot and the likes fall into this complexity.

3.2 Entropy History

Entropy's definition is elusive, statisticians squabble over what entropy means and where its roots were made; here is an attempt to correlate the history.

If you go off looking for information regarding entropy, prepare yourself, it can be overwhelming; you may find yourself staring at thermodynamic equations.

To illustrate my caution, here is an excerpt from an editorial on Diversity and Entropy:

"There are many types of entropy reported in the scientific literature. The great diversity in the concept and definition may cause tremendous problems. My own humble suggestion is the following regarding the main two kinds of entropy: 1. Any information-theoretic entropy (Shannon's entropy, H) should be defined in a way that its relation with information is clear. 2. Any theories regarding thermodynamic entropy (classical entropy, S, or the entropy of Clausius, Gibbs and Boltzmann and Plank) should conform with the second law of thermodynamics."[5]

[5] *Diversity and Entropy- Shu-Kuu Lin – Molecular Diversity Preservation International 1999*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The classical definition of entropy, as theorized by Rudolph Clausius[6] in 1886, was a quantative measure of the amount of thermal energy, not available to do work in thermodynamics, it was to take great leaps to bring it to what we see today. Near the end of the 1800's a group of scientists namely James Maxwell, Ludwig Boltzmann and Josiah Gibbs formed new theory that pushed the ideas of thermodynamic entropy into physics. Utilizing molecular theory it branched entropy into statistical mechanics[7]. It wouldn't be until the work of Claude Shannon was revealed in 1948 that communication & information theory would pave the way to the information-theoretic entropy we see today. Shannon's 1948 paper entitled "A Mathematical Theory of Communication" is ground breaking and very in depth for its day, should you wish a copy it can be found in the archives of bell-labs[8].

3.3 Entropy Defined

Communication theory defines entropy as a numerical measure of the uncertainty of an outcome, utilizing statistical probability. One statistical equation used in determining randomness is the chi-squared equation. Here is a sample equation; you can be the judge if you want to know more about it[9]:

$$E^2 = \frac{\sum_{i=1}^{n} [(y_i - f(x_i))/y_i]^2}{(n-m)} = \frac{\varepsilon^2 X^2}{(n-m)} \rightarrow \varepsilon^2$$

In simpler and more precise terms pertaining to our topic, Entropy is a measure of possible patterns present within random data.

4.0 Applying Entropy to Randomness

Because mathematically generated random data is flawed by obvious discernable patterns, one must find a way to produce randomness at a higher level. This becomes a challenging problem; true random data is difficult to create on a demand basis. The problem forced the emergence of an idea to utilize an "entropy pool" to allow the collection of random data and stir it in occasionally. So where do we get entropy?

---

[6] *On Different Forms of the Fundamental Equations of the Mechanical Theory of Heat Rudolph Clausius 1865*

[7] *Definitions of Entropy – Tim Thompson - 2002*

[8] *A Mathematical Theory of Communications – Shannon*

[9] *Chi-Squared theory*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Typically entropy is collected from either unknown or vastly changing sources. From a computers perspective almost anything going through the processor is predictable, however some very interesting and creative ideas have been thought of for gathering entropy. Following, are some rather notable ones[10]:

Human Interaction

You are human so you must be random, well that isn't exactly true, we all have traits and bad habits. As an example, one form of credential is monitoring the frequency of typing or moving the mouse, this is one type of entropy. You could also use actual typing characters; however there are some distinct patterns, how else would have someone come up with the "dvorak" keyboard layout? Also there is the possibility of leaking information with actual character monitoring. Passwords and random input from users could also be used, but all in all the entropy gained is quite low.

Environmental Gathering

An instance of environment gathering could be using a microphone to collect ambient noise from a room or even a remote location. It is best for only one machine to exist utilizing this form in a given environment, the danger of more is two may collect the same data and provide patterns.

Computer States

The suggestion for utilizing computer states usually focus around use of combining interrupts with microsecond clock data. One interesting idea is to utilize the inherent randomness of the video buffer or possibly the video image changes that appear under the mouse pointer. In addition there are other radical ideas involving the I/O and hard disk, such as measuring turbulence inside a drive that is spinning, that data is currently difficult to collect but all the same it provides entropy.

Intel decided that post 8xx CPU support chips would have a random generator built in allowing for the collection of random data caused by frequency chip noise.

Quantum Events

This is a more difficult source of information, splitting atoms and other quantum type events usually only occur in a controlled scientific research facility (we hope) but in doing research it seems that there are semi-public sites that can be monitored that provide random data utilizing quantum generation.

4.1 Creating better random data

The above shows entropy can be gathered into an "entropy pool" however the question remains, what can be done with all this random data? One answer is to distill it into separate pools making it available for various stages of random data. A negative connotation is that the amount of entropy from these gathered sources could be slow in compiling. To correct for this slowness it is suggested

[10] *Using and Creating Cryptographic-Quality Random Numbers – Jon Callas*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

that the source material could use the pool and select entropy components using a random selection process and then mathematically stir it into the pot resulting in randomly picked data. One accepted way of doing this is using a hash function such as MD5 or SHA.
Simply put, a hash function takes an amount of data pushes it through an algorithm, which yields a numerical fingerprint linking it to the original data. This process in conjunction with the algorithm makes this numerical representation unique. In this way random data can be combined and remain random. There are other ways to distill entropy pools; they range from encrypting it, using a simple XOR function, and utilizing checksums. There is not sufficient room to explore each of these pools but each has advantages and disadvantages. One reason a hash makes such a good distiller is that the result will be different if the order of the inputs is not exactly the same, therefore if an attacker was able to determine which inputs were being used he would still have to figure out the order in which it was hashed, thus creating more entropy[11].

There are other ways to achieve the same results of an entropy pool, one is to use entropy information to "seed" a pseudo-random number generator (PRNG). PRNG's take perceived random sources (usually Machine State or Human) and through cryptographic and mathematical algorithms produce a cryptographic quality random number. Some samples of commercial PRNG's would be Peter Gutmann's PRNG in GUT98 and Colin Plumbs PRNG in PGP.

Bruce Schneier from CounterPane Security has written some absolutely brilliant information available at CounterPane[12]. One such paper written by Bruce Schneier, John Kelsey and Niels Ferguson, describes a PRNG called Yarrow an excellent resource and interesting read pertaining to distilling entropy and utilizing it for randomness.

5.0 Future considerations and Conclusions

Evaluating randomness from how it is utilized within security, defining randomness and its counterpart entropy, identifying ways in getting better random numbers, and describing sources of randomness and entropy, leaves us with a couple of unanswered questions.

One big question is how much entropy does your cryptographic engine use and how many are out there that are based on poor implementations of PRNG's? That is a list that is staggering; though looking at the Yarrow paper leads the writer to believe that some are certainly not acceptable for cryptographic usage.

Another interesting avenue unexplored is vulnerability of PRNG's. There are three known classes of attacks: Direct Cryptographic Attack, Input Based Attacks and State Extension Attacks[12]. Each of these classes depend upon orchestrated

[11] *Using and Creating Cryptographic-Quality Random Numbers – Jon Callas*
[12] *CounterPane Security Publications*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

attacks where attackers could have some level of access to the machine running the PRNG, thereby allowing possible control of inputs and outputs leading to either compromise or failure. There are numerous permutations to these attacks and they tend to be more complex than the scope of this paper. Suffice it to say that there are ways to attack PRNG's and if more detail is desired, it can be found at CounterPane.

The above discussion of TCP sequence numbers creates another question; are they still susceptible and predictable? Over the past few years this has been scrutinized by Operating Systems manufactures and certainly has come a long way since the times of Windows95 and Windows98. Predictions of sequence numbers in earlier Operating systems were described by some security tools as "trivial – Jokes" and because of this many attackers were able to automate man in the middle attacks, examples would be Juggernaut, T-Sight and Hunt[13]. At this point Operating Systems have changed their Internet Protocol stacks to utilize better sources of randomness incorporating in some cases high amounts of entropy, the highest examples of these are OpenBSD (Berkley Software Design) and Linux, Microsoft made some serious changes in Windows2000 and XP however still do not seem to rank among the top network security contenders.

5.1 Looking Forward

Looking forward we can take some solace in reading materials by the masters, that still hold out hope and continually illuminate us with new insights and bolster the information age with security minded intentions. Bruce Schneier's book Secret & Lies contends that the mathematics and technology exist and are being utilized both practically and efficiently by people well educated in their usage. People are fallible and because of this, constant education and heightened awareness could help the quality of cryptosystems development immensely. We are on the brink of new technology such as quantum computing. True randomness is an on going objective and is ripe for evolutionary change, but it must change for network security to stay ahead of those whose objectives it is to do damage, but to what extent that change will be is still too early to tell.

For any wishing further reading include below are some recommended sources for education pertaining to randomness and entropy.

[13] *Man-In-the-Middle Attack - A Brief - Bhavin Bharat Bhansali SANS 2001*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5.2 Additional Resources

Some sources of additional information concerning randomness and entropy:

Applied Cryptography 2nd Edition– Bruce Schneier
Secrets & Lies – Bruce Schneier
Foundations of Cryptography – Oded Goldreich
Secrets of Making and Breaking Codes. - Nickels, Hamilton
Weaknesses in Modern Cryptography – Tim White
A Short Course in Information Theory - David J.C. MacKay
A statistical Test Suite for Random and pseudorandom number generators for Crytpographic applications – NIST special publication 800-22
Internet RFC 1750

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

1. Random Numbers – Russel Kay
   Published Computer World – April 1st, 2002
   http://www.computerworld.com/securitytopics/security/story/0,10801,69677,00.html

2. The Free On-line Dictionary of Computing - Denis Howe
   Source Dictionary.com – 1993 -2001
   http://dictionary.reference.com/search?q=entropy

3. The American Heritage® Dictionary of the English Language, Fourth Edition - Houghton Mifflin Company Source Dictionary.com – 2000
   http://dictionary.reference.com/search?q=entropy

4. Generating a truly Random Number – Leif Svalgaard
   Published Object Z Systems – Cobolreport.com
   http://cobolreport.com/columnists/leif/

5. Diversity and Entropy- Shu-Kuu Lin
   Molecular Diversity Preservation International 1999
   http://www.mdpi.org/entropy/htm/e1010001.htm

6. The Definitions of Entropy - Tim Thompson 2002
   http://www.tim-thompson.com/entropy1.html

7. A Mathematical Theory of Communications – Claude Shannon
   Bell System Technical Journal 1948
   http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html

8. Chi-Squared Theory
   Research Solutions & Resources – 2001
   http://www.consultrsr.com/resources/chisquared.htm

9. Using and Creating Cryptographic-Quality Random Numbers – Jon Callas(1996)
   http://www.merrymeet.com/jon/RTFToC1

10. CounterPane Labs Security Publications 1999 – 2000
    http://www.counterpane.com/yarrow-notes.html
    http://www.counterpane.com/prf-prp.html
    http://www.counterpane.com/pseudorandom_number.html

11. Man-In-the-Middle Attack - A Brief - Bhavin Bharat Bhansali
    SANS Infosec Reading Room – 2001
    http://www.sans.org/rr/threats/middle.htm

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 5th, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
Security Leadership Summit & Training	Dallas, TXUS	Dec 03, 2015 - Dec 10, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced