SANS Institute
InfoSec Reading Room

Rationally Opting for the Insecure Alternative: Negative Externalities and the Selection of Security Controls

As with all aspects of business and the economy, information security is an economic function. Security can be modeled as a maintenance or insurance cost as a relative function but never in absolute terms. As such, security can be seen as a cost function that leads to the prevention of loss, but not one that can create gains (or profit). With the role of a capital investment to provide a return on investment, security is a defense against unforeseen losses that cost capital and reduce profitability.

Rationally opting for the insecure alternative: Negative externalities and the selection of security controls

GIAC (GSLC) Gold Certification

Author: Craig S Wright, craig.wright@information-defense.com
Advisor: Tim Proffitt

Accepted: Sept 05th 2011

Abstract

As with all aspects of business and the economy, information security is an economic function. Security can be modeled as a maintenance or insurance cost as a relative function but never in absolute terms. As such, security can be seen as a cost function that leads to the prevention of loss, but not one that can create gains (or profit). With the role of a capital investment to provide a return on investment, security is a defense against unforeseen losses that cost capital and reduce profitability.

1. Introduction

Absolute security does not exist and nor can it be achieved. The statement that a computer is either secure or not is logically falsifiable (Peisert & Bishop, 2007), all systems exhibit a level of insecurity. An attacker with sufficient resources can always bypass controls. The goal is to ensure that the economic constraints placed upon the attacker exceed the perceived benefits to the attacker. This generates a measure of relative system security in place of the unachievable absolute security paradigm that necessarily results in a misallocation of resources.

Relative computer security can be measured using six factors (Aycock, 2009):

1. *What is the importance of the information or resource being protected?*
2. *What is the potential impact, if the security is breached?*
3. *Who is the attacker likely to be?*
4. *What are the skills and resources available to an attacker?*
5. *What constraints are imposed by legitimate usage?*
6. *What resources are available to implement security?*

The result is that security is a relative risk measure related to organizational economics at the micro level and the economics of national security toward the macro level. This consequentially leads to a measure of security in terms of one's neighbor. The question is not, "*am I secure*", but rather, "*am I more secure than my neighbor*?"

Multiple means of assessment are possible. Any other system is your neighbor on the Internet when viewed from the perspective of a Worm. Conversely, targeted attacks have a purpose. Neighbors may be other government systems, critical infrastructure, and a class of companies or an industry sector. In each instance, achieving a measure of security relates to a set of relative terms.

For all the changes in outward trappings and fashion many of our values are still the same. This statement, true in business as a whole is particularly true in the realm of information security. Our psychology has not varied and new security leaders still make

the mistakes of the past "*which always causes a new prince to burden those who have submitted to him with his soldiery and with infinite other hardships which he must put upon his new acquisition*" (Machiavelli, Il principe, 1513). In information security, we also make the same mistakes; we seek an absolute security that can never be obtained and fail to see that relative security can provide many solutions that cannot otherwise be obtained.

In seeking to measure relative security, we can look to the cost of alternatives and compare the options we have available to see the best solutions when we are constrained by a given set of limits. It is to be remembered is that we are always restricted in some manner and that economics is a hard and fast barrier stopping us from even approaching an absolute level of security let alone achieving it. Only in weighing and measuring options can we allocate resources most efficiently and hence achieve the maximum level of achievable security for our systems.

2. Assessing Individual Security Costs

The most effective security solution is that which provides the best level (an optimised state) for "the least cost". Costs to the consumer are minimised at the point where security costs exactly equal the expected loss that is associated with the risk function.

More security costs = higher costs to the consumer.

Higher expected loss from risk = higher costs to the consumer.

One expects that as expenditure on security decreases the expected loss, the costs to the consumer be minimised where the additional expenditure of $1 on security reduces the expected risk based loss by exactly $1.

Security is a cost function. Business necessarily passes this cost onto the consumer where possible. This cost will be passed in the event that the business can do so and still retain profitability. Where a business cannot pass the cost, it can result in reduced profit and this occurs directly where alternatives exist (this is the product is

elastic or consumers are willing to reduce their use if costs increase). We can express the expected cost formula for the supply of these types of services against a loss function as:

$$C_s = D(x,y) + x + y \tag{1}$$

In (1), the loss function $D(x,y)$ and the damage to x (the producer) and y (the consumer) are modelled arithmetically. As in all areas of economics, the marginal gains in D_x offset those of D_y.

In these calculations, calculations, $D_{xy}D_{xy} > D_{xx}D_{yy}$, which creates the inference that the inputs are substitutes. As the producer spends more on security, the consumer spends less and vice versa. The exact composition of these values varies based on the nature of the product with elastic supply affected more than an inelastic supply.

The real issue and goal in security becomes the creation of Cournot-Nash equilibria. This is an outcome where X_e and Y_e together form a Cournot-Nash equilibria for a given value of Y_e (designated as y_e); the x which maximises X's utility is X_e and given X_e that y which maximises Y's utility is y_e. This does not require that the equilibria be Pareto optimal.

At present, the cost functions directed towards many industries (such as banks in regulated countries including Australia) are sufficient in that there is but a trivial increase in marginal demand for the consumer for an incremental increase in security expenditure. The producing company is likely to do little and that which they do conduct has a minimal effect. For instance, Microsoft is unlikely to greatly improve the security of its operating system through minimising patches due to the increasing cost of finding additional bugs in its software. If it did so, the cost point is such that Microsoft's profit diminishes. Consumers are generally unwilling to bear the cost increment that this would entail. The incremental cost of finding additional bugs exceeds the total cost to all

consumers of taking an alternative course of action (such as installing HIDS[1] and Host firewalls).

The loss for the consumer lessens to a lower extent than the loss of the producer. With fraud loss limits of $50 in countries such as Australia for online transactions, banks in these locations have an incentive to minimise the loss to the consumer. Perversely, this can incentivise the consumer against adequately securing their system. If the consumer expects to lose a maximum of L_{iy} (which is set at $50 for credit card transaction fraud in Australia) for any given incident (i) where the total expected damage is

$$D_y = \sum_{i=1}^{n} L_{iy} \qquad D_x = \sum_{i=1}^{n} L_{ix}. \tag{2}$$

The expected annual number of incidents per consumer, n, derives from the total number of incidents that have occurred divided by the total number of consumers of a class (i.e. the total pool of credit card users).

$$E(n) = \frac{\#incidents}{\#consumers} \tag{3}$$

Setting C_{Ty} as the total cost to the consumer of implementing controls, if the expected total loss to the consumer $D_y < C_{Ty}$, it is doubtful that the consumer will pay for additional protection. For instance, if a high-end HIDS and anti-malware product costs $C_{Ty} = \$225$, and the consumer experiences n=4 incidents in a usual year, the expected damage $D_y = \sum_{i=1}^{n} L_{iy} = \$200.$ As $D_y < C_{Ty}$, it is not in the interest of the consumer to adequately protect their system. The user of a system that requires more security then the mean level of control provided by a vendor can implement increased security controls on their system, but this would either require that the consumer experience other measurable losses or that $D_y > C_{Ty}$ for this consumer.

[1] Host Intrusion Detection Software.

Here we see that the anti-fraud efforts by banks and credit card companies create a negative incentive to consumers. The loss to the vendor L_{ix} currently averages $237 (Ben-Itzhak, 2009) for each lost set of credentials. The result is that it is in the interest of the financial company to provide the consumer with a compensating control. Holding the consumer liable if they had failed to use the enhanced controls over security would result in $D_y > C_{Ty}$ and hence an incentive for the consumer to protect their system.

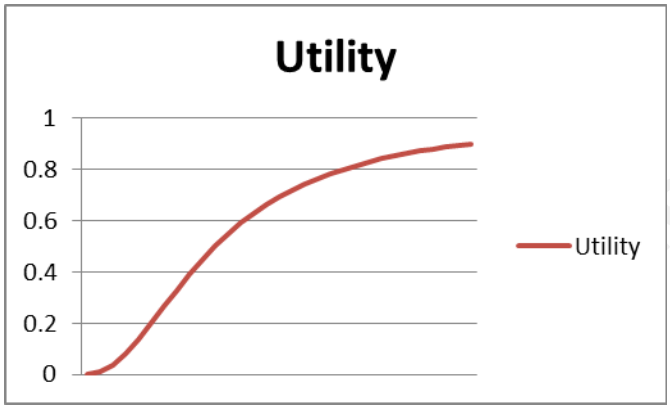

Fig. 1. Modeling software security.

Capital invested by the consumer in securing their system has a greater marginal effect than that of the producer in the case of an organisation such as Microsoft. A consumer can purchase HIDS and host firewall software for less than the cost that it would cost Microsoft to perfect their software through formal verification and hence remove more bugs.

The expected damage, $E(Damage)_i = P(x_{ai}).D_{Tot}$ or the expected damage is equal to the probability of a breach times the amount of damage suffered in a breach. We express this function for each user or as a total cost function for all users, $E(Damage) = \sum_i \left(P(x_{ai}).D_{Tot} \right)$. Here we can clearly see that the total amount of damage is a function of not only the producer, but also the consumer. The optimal solution is to find a point that minimises the total costs. This is the expected damage as a loss function plus the costs of damage prevention of a compromise of other loss. The

damage is expressible as a function of the producer and consumer (user) costs,

$$C_T = Cost_{Tot} = \sum_i \left[P(x_{ai}) D(x_{ai}) \right] + C_v + \sum_i \left[C_u(i) \right] \quad (4)$$

The first order conditions are

$$P'(x_{ai}) D(x_{ai}) + 1 = 0 \quad (5)$$

$$D'(x_{ai}) P(x_{ai}) + 1 = 0 \quad (6)$$

That is, the user should increase his expenditure on precaution (preventing a breach) until the last dollar spent on precaution by the user reduces the expected damage by $1. Moreover, the producer should increase her expenditure on reducing the possible damage in case of a breach until the last dollar spent on precaution by the producer reduces the expected damages by $1.

Clearly, the greater the likelihood of the user experiencing a breach, or the larger $P(x_{ai})$ is for the user, the greater the precaution that they should undertake. In the case of a producer who is a software vendor, they will (generally) sell their products to a wide range of users with varying levels of likelihood that each will experience a breach. That is, the software vendor is acting with imperfect information.

We denote the optimal amount of precaution is the solutions to Equations (5) and (6) using the expressions, C_v^{Ω}, $C_u^{\Omega}(i)$ and where the total costs for all users is optimised at $\sum_i \left[C_u^{\Omega}(i) \right]$.

The marginal utility expenditure of security means that the value of security decreases the more we add. There is reason for this. If we spend more than the value of the organisations capital, it is simple to see that the producer will not survive long. It is more than this, we only need to reduce profitability for a producer to fail, not the capital.

The level of damages suffered by a user depends on both the pre-breach behaviour of the user and the vendor. The vendor is in a position where reputation influences sales (demand) and hence the willingness to add layers of testing and additional controls (all of which increase the cost of the software). As the market for

software varies in its elasticity (Stolpe, 2000) from the highly inelastic in small markets with few competitors (e.g. Electricity markets) to highly elastic (e.g. Operating Systems), the user has the ability to best determine their needs. The user may select customised software with warranties designed to reduce the levels of breach that can occur[2]. This comes with an increased cost.

Software vendors (unless specifically contracted otherwise) do not face strict liability for the damage associated with a breach due to a software vulnerability (Hahn & Layne-Farrar, 2007; Scott, 2007). Although some desire the introduction of negligence rules for software vendors (Scott, 2007), this creates a sub-optimal outcome. The user can (excepting certain isolated instances);

1. select different products with an expectation of increased security[3] (Devanbu & Stubblebine 2000),
2. add external controls (through the introduction of external devices, create additional controls or use other software that enhances the ability of the primary product),
3. Increase monitoring for attacks that may be associated with the potentially vulnerable services (such as by the use of an IDS[4]).

By limiting the scope of the user's responsibility[5], the user's incentive to protect their systems is also limited (Hahn & Layne-Farrar, 2007). That is the user does not have the requisite incentive to take the optimal level of precautions. Zero day attacks (DShield, 2010) are not the cause of the majority of breaches. When patches already exist for known vulnerabilities that could lead to a breach, users will act in a manner (rational behaviour) that they expect to minimise their costs (White & Stjohn, 2006). Whether risk seeking or risk adverse, the user aims to minimise the costs that they will experience. This leads to a wide range of behaviour with risk adverse users taking additional

[2] This is the case in many of the former "Rainbow Book" A1 level software installs used in military installations
[3] The reputation of a secure product adds value or can conversely negate value if the product is seen as being insecure.
[4] Intrusion detection system (IDS).
[5] Strict liability for software vendors limits the user choice as the user expects the vendor to act and does not take the appropriate level of care.

precautions and risk neutral users can accept their risk by minimising their upfront costs (which may lead to an increase in loss later).

In any event, the software vendor as the cause of a breach is not liable for any consequential damages[6]. This places the appropriate incentives on the user to mitigate the risk. At the same time, the vendor has a reputational incentive to minimise the risk to their reputation. This was seen a number of years ago where the costs of bugs to the consumer from Microsoft was deemed as being exceedingly high. The vendor response was to change their coding practices and to reduce the number of vulnerabilities in their released code significantly.

2.1. Rational Limits to Security

We can apply rational choice models to security. In this, security is the composite good that we seek to model. As we move the security expenditure from a lower to a higher value, the returns on that expenditure increase to a maxima and then decrease.

The optimal point is one where security expenditure and expected returns result in positive growth. As expenditure on security increases, the expenditure increases, but the return approaches zero. That is, for each additional dollar spent, the return tends to zero.

The logic for this is simple. If we have a capital investment of I, as the amount spent on security approaches I, the amount of capital and hence the investment that remains tends to zero. There is more to this however. The value that matters is not the capital, but rather the expected return on capital. Investors expect a return on their money. If the return is less than the risk associated with the investment, the investor withdraws invested capital and the company will fold. Investment in security lowers the chance of loss; it does not create profit in itself.

The maximum loss is limited as the capital in a firm is recoverable within set finite bounds. The maximum expenditure on security is not in theory limited, although practically there are limits to the amount of capital that acquirable in any investment

[6] For instance, legal rules when a part is not supplied do not leave the supplier liable for the lost profits due to the unavailability of the part. This rule is ensconced in *Hadley v. Baxendale, 156 Eng. Rep. 145 (1854)*.

cycle. In any event, security expenditure has failed where it costs more than it expects to save.

3. Assessing Security

Being a relative function, not only does the profitability of an individual class (be that organization, group or nation) factor into the calculation of security risk, but the relation to a classes neighbors also needs to be measured.

The cost function is in the criminals favor without additional input from the consumer. There is no impetuous for the bank to move to a more secure (and also more costly) means of protecting consumers when the criminal can still gain access to the consumer's system. One part of the problem is the regulations that plague banking. The requirement to authenticate customers when calling for their privacy makes it simple for a criminal to pose as the bank and obtain the information. So even if a more secure means is selected, it is trivial to bypass many controls using social engineering and other less technical methods.

Whilst there are greater losses from consumer inaction than supplier inaction, the consumer's failure to secure their system and refrain from the use of systems at insecure locations all compound to make it more likely to have a loss though this means.

At all points of an assessment, we have to also take the time value of money into account. The value of capital is not set and fluctuates with time. To evaluate costs, we need to take both cost and the point at which the cost is expressed into account.

3.1. Economic Equivalence

The comparison of the financial characteristics of the alternatives is on an equivalent basis in order to compare any set of two or more alternatives. Two options are equivalent when they have the same effect. Monetary values are termed as equivalent when they have the same exchange value. We define this as:

1. The comparative amount of each monetary sum,
2. The times of the occurrence of the sums can be aligned.

3. An interest rate can be used to compare differences in the time of payment.

The general equivalence function is:

$$\text{PE, AE or FE} = f(F_i, i, n) \quad (7)$$

This equation holds for values of t between 0 and n. The equivalence equation uses:

F_t = the rate of monetary flow at the end of time period t.

i = the rate of interest for the time period.

n = the number of discrete time periods.

The security and risk product lifecycle defines the function of the acquisition and utilisation phases. A system with a longer MTBF[7] has a greater return on the initial investment. Similarly, larger upfront investments in security reduce the amount of capital available for investment. The financial present equivalent function [PE(i)] is a value calculation that relates to the difference between the present equivalent capital value and the present equivalent costs for a given alternative at a given interest rate.

The present equivalent value at interest rate i over a total of n years is stated as:

$$PE(i) = F_0(^{P/F,i,0}) + F_1(^{P/F,i,1}) + \ldots + F_n(^{P/F,i,n})$$
$$= \sum_{t=0}^{n} F_t(^{P/F,i,t}) \quad (8)$$

4. Stag Hunt

George Akerlof's model, "*A Market for Lemons*" (Akerlof, 1970) as was designed for modeling quality uncertainty has been proposed as a game model for the software industry (Anderson, 2001). This model is based on information asymmetry and the presumption that the vendor has more knowledge of the product than the user. This is a fallacy in that the software vendor is incentivized to correct bugs as early in the process as is possible (the later a bug is discovered in the development process, the more it costs

[7] Mean time between failure

to fix). Hence, the vendor does not have more of an idea of the expectations of flaws than a knowledgeable user. Further, the user knows how they plan to deploy the software; the vendor does not have this information and may have little insight into what other interactions may occur.

The software game is also sequential with multiple iterations. The vendor wants to maintain a relationship with the user and as the software is used, it can be assessed against the assertions of the vendor. Further, the user can generally compare the past performance of the vendor.

A better game model for the software industry is the "*Stag Hunt*". This was based on Jean Jacques Rousseau's postulations of a co-operation strategy between two hunters (Skyrms, 2004). These individuals can either jointly hunt a stag or individually hunt a rabbit. The largest payoff is assigned against the capture of a stag which provides a larger return than the hare. The hunting of a stag is more demanding and requires mutual cooperation. If either player hunts a stag alone, the chance of success is negligible and sub-optimal. Hunting stags is most beneficial for society in that this activity creates the optimal returns. The problem with this game is that it requires a lot of trust among the players.

		User	
		Create Secure Software	**Add Features**
Software Vendor	**Create Secure Software**	10, 10 A, W	1, 7 B, X
	Add Features	7, 1 C, Y	5, 5 D, Z

Fig. 2. Software Markets as a "Stag Hunt".

This game has two pure strategy equilibria in which both of the players prefer the lower risk equilibrium to the higher payoff equilibrium. The game is both *Pareto optimal* and *Hicks optimal*, but the sub-optimal and hence inefficient equilibrium poses a lower risk to either player. As the payoff variance over the other player's strategies is less than that of the optimal solution, it is more likely that this option will be selected. Another

way of stating this is that the equilibrium is payoff-dominant while the other strategy is risk-dominant.

The strategy between the vendor and the software user is displayed in Fig 2. In this, the numerical representations represent the payoff figures for the specific case (the software market) and the generalized relations take the form:

$$\begin{aligned} A > C \geq D > B \\ W > X \geq Z > Y \end{aligned} \tag{9}$$

The outcomes are not definitive statements of what will be produced. In this game, the "*Stag*" is a desire to "*Create Secure Software*" and the "*Hare*" the fallback to adding more features. A desire is not a case of creating fewer bugs by itself, but rather a combination of adding controls and testing to software. Such an example would be the addition of the SP2 to Windows XP by Microsoft. Additional testing is effective to a point and more can be done than is occurring at present.

The payoffs for creating more secure software are great for both the vendor and the user, but the risk of a misaligned strategy leads to the sub-optimal equilibria. What is needed is a signaling process. A signal will allow the players to align to the more optimal strategy. It is not only in the user's interest to have more secure software, but also is in the interest of the vendor. Patching is expensive and the vendor can reasonably charge more for secure software.

A problem with a sub-optimal equilibrium is that "*talk is cheap*". A player's strategy is not only whether to hunt stag or hare, but also what signal to send, and how to respond to signals he receives. In order to switch from the hare hunting equilibrium (more Features) to the other, over three quarters of the population must simultaneously switch strategy to require secure software. This is a simple situation when there are only two (2) players, but becomes more complex in an n-player game.

As the ratio between the payoff for stag hunting and the payoff for hare hunting is reduced, the incentives to move towards stag hunting decreases. As a result, it becomes less likely that software security will be made into a primary goal of either party. As such, where the introduction of features and the "*new killer app*" occur more frequently,

software security lags and it becomes more likely that a change from a stag hunting equilibrium to a hare hunting equilibrium will occur. It is hence less probable that an alteration of the players strategy from hare to stag.

Since neither player has an incentive to deviate, this probability distribution over the strategies is known as a correlated equilibrium of the game. Notably, the expected payoff for this equilibrium is $7(1/3) + 2(1/3) + 6(1/3) = 5$ which is higher than the expected payoff of the mixed strategy Nash equilibrium.

5. Conclusion

The addition of measures that take externalities into account act as a signaling instrument that reduce information asymmetry and improve the overall risk position of both the consumer and vendor. The development of a software risk derivative mechanism would be beneficial to security (Jaziar, 2007) through the provision of a signaling process to security and risk.

In economic terms, we want to assign liability such that the optimal damage mitigation strategy occurs. The victim will mitigate their damages where no damages for breach need apply in respect of the optimal strategy and payoffs. The rule that creates the best incentives for both parties is the doctrine of avoidable consequences (marginal costs liability).

Mitigation of damages is concerned with both the post-breach behaviors of the victim and the actions of the party to minimize the impact of a breach. In a software parlays', this would incur costs to the user of the software in order to adequately secure their systems. This again is a trade-off. Before the breach (through software failures and vulnerabilities that can lead to a violation of a system's security) the user has an obligation to install and maintain the system in a secure state.

Blanchard & Fabrycky (2006) note:

> "*When the producer is not the consumer, it is less likely that potential operation problems will be addressed during development. Undesirable outcomes too often end up as problems for the user of the product instead of the producer."*

Both software and Security in general have an "*agency problem*". Only by measuring and reporting these costs in financial and economic values can we start to truly fix the problems.

Some (Belovich, 2010) have stated that as a stochastic process, risk cannot be predicted. This comes down to the fallacy of absolute risk. In this, they have misunderstood the nature of dynamical systems (of which risk is but one example) and have confused random events with chaotic systems. In a chaotic system, the next state is not random, but is probabilistically related to the last state. Stochastic systems have provided us with rich models allowing for the growth in many communications systems and the creation of switched digital voice networks. Correctly deployed, they can also aid us in minimizing the impacts of risk.

Consequently, in knowing the relative risk at a point in time we can make informed decisions that minimize potential losses when comparing alternate solutions. It is these alternate solutions that need to be measured whether these involve adding new controls or simply doing nothing and investing the money saved. What matters is the best economic return. The unswerving clinging to old values (including the false notion of absolute security) is not conducive to minimizing risk. Rather those who embrace and manage risk forge ahead in the wake of those who embrace their former models.

Absolute security does not exist and nor can it be achieved. The statement that a computer is either secure or not is logically falsifiable as all systems exhibit a level of insecurity and an attacker with sufficient resources can always bypass controls. The goal is to ensure that the economic constraints placed upon the attacker exceed the perceived benefits to the attacker and to do this, we need to be able to create measures.

These measures do not actually quantify the level of security a system has, but the relative security of the system in one state when compared to another. It is the difference that matters. In measuring relative security instead of seeking an unachievable state of an absolute that can never be reached, we can better allocate scarce resources where they will have the maximum benefit and avoid the pitfalls of attempting to achieve the unachievable absolute security paradigm that necessarily results in a misallocation of resources.

Craig S Wright, Craig.Wright@information –defense.com

In this paper, we have endeavored to introduce a few techniques that can be used to model and hence measure risk. Remember, risk and security are relative and never absolute values. The best we can do is to measure one state against an alternative state and see which is best.

Craig S Wright, Craig.Wright@information –defense.com

6. References

Akerlof, George A. (1970). "The Market for 'Lemons': Quality Uncertainty and the Market Mechanism". Quarterly Journal of Economics 84 (3): 488–500. doi:10.2307/1879431

Anderson, Ross. 2001. "Why Information Security is Hard – an Economics Perspective." 17th Annual Computer Security Applications Conference. New Orleans, LA, USA.

Aycock John Computer (2006) "Viruses and Malware", Advances in Information Security, Vol. 22, Springer US.

Bejtlich, Richard (2010) "Thor vs Clown" Tao Security, http://taosecurity.blogspot.com/2010/02/thor-vs-clown.html

Ben-Itzhak, Y. (2009). "Organised cybercrime and payment cards." Card Technology Today 21(2): 10-11.

Blanchard, BS & Fabrycky, WJ (2006) Systems Engineering and Analysis, 4th Edition Pearson Prentice Hall, Upper Saddle River, NJ, USA

Devanbu, P. T. and S. Stubblebine (2000). Software engineering for security: a roadmap. Proceedings of the Conference on The Future of Software Engineering. Limerick, Ireland, ACM.

DShield. http://www.dshield.org, 2006-2010.

Hahn, Robert W.; Layne-Farrar, Anne, "The Law and Economics of Software Security", Harv. J.L. & Pub. Pol'y 283 (2006-2007)

Jaziar, R. (2007). Understanding Hidden Information Security Threats: The Vulnerability Black Market. Paper presented at the 40th Annual Hawaii International Conference on System Sciences (HICSS'07).

Machiavelli, Niccolò (1513) "Il principe" (The Prince)

Peisert, S. and Bishop, M. How to Design Computer Security Experiments WG 11.8 International Federation of Information Processing, Springer Boston, 2007.

Scott, Michael D "Tort Liability for Vendors of Insecure Software: Has the Time Finally Come";. 67 Md. L. Rev. 425 (2007-2008)

Craig S Wright, Craig.Wright@information –defense.com

Skyrms, Brian (2004) "The Stag Hunt and the Evolution of Social Structure" Cambridge University Press

Spitzner, L. Know your enemy: Honeynets. http://project.honeynet.org/papers/honeynet, 2005.

Stolpe, M. (2000). "Protection Against Software Piracy: A Study Of Technology Adoption For The Enforcement Of Intellectual Property Rights." Economics of Innovation and New Technology 9(1): 25-52.

Vinod Yegneswaran , Paul Barford , Johannes Ullrich, Internet intrusions: global characteristics and prevalence, Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, June 11-14, 2003, San Diego, CA, USA

White, Dominic Stjohn Dolin (2006) "Limiting Vulnerability Exposure through effective Patch Management: threat mitigation through vulnerability remediation" MASTER OF SCIENCE Thesis, Department of Computer Science, Rhodes

Last Updated: July 20th, 2018

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Riyadh July 2018	Riyadh, SA	Jul 28, 2018 - Aug 02, 2018	Live Event
SANS Pittsburgh 2018	Pittsburgh, PAUS	Jul 30, 2018 - Aug 04, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LAUS	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, IN	Aug 06, 2018 - Aug 11, 2018	Live Event
Security Awareness Summit & Training 2018	Charleston, SCUS	Aug 06, 2018 - Aug 15, 2018	Live Event
SANS Boston Summer 2018	Boston, MAUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS San Antonio 2018	San Antonio, TXUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS August Sydney 2018	Sydney, AU	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS New York City Summer 2018	New York City, NYUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Northern Virginia- Alexandria 2018	Alexandria, VAUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Krakow 2018	Krakow, PL	Aug 20, 2018 - Aug 25, 2018	Live Event
Data Breach Summit & Training 2018	New York City, NYUS	Aug 20, 2018 - Aug 27, 2018	Live Event
SANS Chicago 2018	Chicago, ILUS	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Prague 2018	Prague, CZ	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Virginia Beach 2018	Virginia Beach, VAUS	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS San Francisco Summer 2018	San Francisco, CAUS	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS Copenhagen August 2018	Copenhagen, DK	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS SEC504 @ Bangalore 2018	Bangalore, IN	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS Wellington 2018	Wellington, NZ	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, NL	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, JP	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FLUS	Sep 04, 2018 - Sep 09, 2018	Live Event
SANS MGT516 Beta One 2018	Arlington, VAUS	Sep 04, 2018 - Sep 08, 2018	Live Event
Threat Hunting & Incident Response Summit & Training 2018	New Orleans, LAUS	Sep 06, 2018 - Sep 13, 2018	Live Event
SANS Baltimore Fall 2018	Baltimore, MDUS	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS Alaska Summit & Training 2018	Anchorage, AKUS	Sep 10, 2018 - Sep 15, 2018	Live Event
SANS Munich September 2018	Munich, DE	Sep 16, 2018 - Sep 22, 2018	Live Event
SANS London September 2018	London, GB	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NVUS	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS DFIR Prague Summit & Training 2018	Prague, CZ	Oct 01, 2018 - Oct 07, 2018	Live Event
Oil & Gas Cybersecurity Summit & Training 2018	Houston, TXUS	Oct 01, 2018 - Oct 06, 2018	Live Event
SANS Brussels October 2018	Brussels, BE	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Pen Test Berlin 2018	OnlineDE	Jul 23, 2018 - Jul 28, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced