

Guide

Jericho Forum[®] Self-Assessment Scheme



Copyright © 2010, The Open Group

Licensed under Creative Commons Attribution-No Derivative Works 2.0 UK:

England & Wales – <http://creativecommons.org/licenses/by-nd/2.0/uk/>

You are free to copy, distribute, display, and perform the work, subject to appropriate attribution (the Jericho Forum), except that in addition you may make derivative works, providing such works do not claim to be endorsed by the Jericho Forum.

This document has not been verified for avoidance of possible third-party proprietary rights. In implementing this document, usual procedures to ensure the respect of possible third-party intellectual property rights should be followed.

Guide

Jericho Forum® Self-Assessment Scheme

Published by The Open Group, March 2010.

Comments relating to the material contained in this document may be submitted to:

The Open Group
Thames Tower
37-45 Station Road
Reading
Berkshire, RG1 1LX
United Kingdom

or by electronic mail to:

jerichoforum-interest@opengroup.org

Contents

1	Background and Rationale	1
1.1	Why this Self-Assessment Scheme?.....	1
1.2	Value-Add: Why will a Vendor or Customer use this Self-Assessment Scheme?	2
1.3	Self-Policing	3
1.4	Initial Industry Response	3
2	Using the Self-Assessment Scheme	5
2.1	Caveats	5
2.2	How to Use this Document	5
2.3	How to Complete a Self-Assessment	6
3	Self-Assessment Template	7
4	Self-Assessment Scorecard	31

Preface

The Jericho Forum

The Jericho Forum is a Managed Consortium of The Open Group. Founded in January 2004, the Jericho Forum is an international IT security thought-leadership group dedicated to defining ways to deliver effective IT security solutions that will match increasing business demands for secure IT operations in our open, Internet-driven, globally networked world. Its members include multi-national corporate user organizations, major security vendors, solutions providers, and academics, all working together to:

- Drive and influence development of secure architectures, technology solutions, and implementation approaches, for securing our de-perimeterizing IT world, to enable safe, secure collaborative interworking, globally between enterprises – business partners, customers, suppliers, and out-workers.
- Support development of open standards that will underpin these solutions.

By 2008, the Jericho Forum had raised industry awareness of the major information security challenges that increasing erosion of corporate boundaries was creating, published its commandments (design principles) on requirements for effective information security in de-perimeterized environments. It also published a number of requirements papers explaining de-perimeterization and effective security responses. In these papers, the requirement for information-centric security – moving the protection close to the data asset – is a key objective. In January 2009, the Jericho Forum published its Collaboration Oriented Architectures (COA) Framework for developing secure architectures for de-perimeterized environments.

Current activities are focused on maturing the COA Framework solution space, and analyzing requirements for enabling secure business collaboration in the de-perimeterized environment that Cloud Computing represents. In April 2009, the Jericho Forum published its Cloud Cube Model, which assessed the Cloud Computing space from the viewpoint of a business manager assessing the risks and benefits of extending business operations and collaborative partnerships into different types of cloud. It is building on this with developments in several related areas, one in particular being Identity and Access Management, as part of the essential security infrastructure for managing business collaborations in de-perimeterized environments. It has established its own collaborative partnerships with other expert groups – including the Cloud Security Alliance – to share knowledge and experience in influencing development of effective information security solutions.

The Jericho Forum is now recognized in the industry as the visionary thought-leadership group it set out to become, pointing the way forward on the security solutions that IT-dependent organizations need and want to buy to secure their business operations in the future.

For more information, see www.jerichoforum.org.

This Document

In 2006, the Jericho Forum published its commandments (design principles) for effective security in de-perimeterized environments). This Self-Assessment Scheme takes each commandment in

turn, and asks “nasty” (i.e., difficult, searching, probing) questions to reveal how effectively a given information security product or solution meets the criteria implicit in that commandment. It then gathers the answers into a Self-Assessment Scorecard.

The “self-assessment” aspect of this scheme is important – it is a self-policing scheme to ensure it is low-cost and low-maintenance for the Jericho Forum, while representing high value to product suppliers who wish to show how well they satisfy our commandments, to customers who wish to check how well a product or solution meets their requirements, and to IT architects and designers who want to check how secure their designs are.

Trademarks

Boundaryless Information Flow™ and TOGAF™ are trademarks and Making Standards Work®, The Open Group®, UNIX®, and the “X” device are registered trademarks of The Open Group in the United States and other countries.

Jericho Forum® is a registered trademark of the Jericho Forum.

The Open Group acknowledges that there may be other brand, company, and product names used in this document that may be covered by trademark protection and advises the reader to verify them independently.

Acknowledgements

The Jericho Forum gratefully acknowledges the contribution of Paul Simmonds (Founding Member & Board Member of the Jericho Forum) as project leader, chief contributor, and editor of this Self-Assessment Scheme. It also acknowledges the significant contributions by Andrew Yeomans (Founding Member & Board Member) and Jamie Bodley-Scott (Member) in the development of this document, and the contributions of other members in extensive review and feedback during its development.

1 Background and Rationale

1.1 Why this Self-Assessment Scheme?

The Jericho Forum Self-Assessment Scheme has been described as “the set of nasty questions to ask your security vendors”, to check whether they provide the security solutions you need, and expose shortcomings in the features they may be claiming their offerings provide. In this context, “nasty” means searching, probing, difficult – raising issues that are critical to providing effective security in de-perimeterized environments.

In 2006, the Jericho Forum published its commandments¹ (with a minor revision in May 2007). Whilst building on “good security”, these commandments specifically address those areas of security that are necessary to achieve secure operation in environments where corporate boundaries are increasingly being undermined – i.e., becoming de-perimeterized – with perimeter firewalls being bypassed by VPN tunnels, wireless/mobile, etc. These commandments have been adopted as design principles by many IT architects and designers. They continue to serve as a set of benchmarks by which the effectiveness of information security concepts, solutions, standards, and systems can be assessed and measured.

The Jericho Forum has known for a long time that enlightened organizations have been using the Jericho Forum commandments as part of their Request for Procurement (RFP) processes with vendors, and also that system and security architects use them to evaluate the effectiveness of their security designs. The reason why is that the commandments raise those searching issues that customers need to ask their vendors about how effectively a security product or solution will perform in their de-perimeterizing environments, and that system architects and designers find valuable in evaluating the security of their designs.

Over this time, the Jericho Forum has been pressed by the buy-side of our IT industry to “tell us the searching questions we should ask our vendors so we can judge how well their products meet our security needs”. In the April 2008 Infosecurity Europe event we did just that – we ran a “chalk & talk” session in which we presented some of the IT security architectures you might wish to use, explained how these operate in perimeterized environments, and then how they could be re-designed to provide the same functionality but as if you were outside your corporate perimeter (i.e., de-perimeterized). Effective security should work to the same effect whether you are inside your corporate perimeter (perimeterized) or outside it (de-perimeterized). Unsurprisingly, out of this came a set of “nasty” questions that we invited our audience to ask their vendors so they could differentiate which competing products performed best for their needs. Feedback from that session was so encouraging that we decided to formalize it into a set of “searching questions to ask your vendor”, which we present in this document.

So, in this Self-Assessment Scheme, for each of our 11 commandments, we present a set of questions which aim to bring out answers indicating how well a security product satisfies each commandment. The answers can then be compiled into a Self-Assessment Scorecard that can be used by vendors, customers, and system architects/designers alike:

¹ Jericho Forum Commandments: published by the Jericho Forum and freely available for viewing and download from www.opengroup.org/jericho/commandments_v1.2.pdf.

- Vendors can self-assess their product and may then choose to use their Self-Assessment Scorecard in responses to Requests for Quotation (RFQs), to indicate how “ready” their product is.
- Customers – especially middle managers and buyers from the many small and medium business enterprises who have not been closely involved in our development of these commandments, and who do not have the depth of IT security expertise that larger corporations have – can ask their vendors the “nasty” questions, and thereby make their own assessment of how each product performs against their requirements.
- User organizations can apply the Self-Assessment Scheme to their own IT system implementations and architectures to assess how secure they are.
- System Architects can use them to evaluate the effectiveness of the security aspects in their designs.

Good examples always help to bring a scheme like this alive. So, for example, one of the commandments is that you should not use inherently insecure protocols. In other words, you should not be using telnet; instead you should use the secure version of telnet; i.e., SSH. Similarly, you should not use FTP but instead use SFTP – which is secure. So the Jericho Forum has said from the start “use secure protocols”. The self-assessment question here, therefore, to score “Good” asks:

- Out-of-the-box are you using secure protocols?
- Are they documented?
- Have you explained that if you give the option to downgrade (e.g., for a web server, out-of-the-box it is https, but there’s an option to downgrade to the insecure version – http), what are the pros and cons of downgrading to the less secure protocol?

We appreciate this requires additional work by the vendor, but it is so important to insist that vendors configure their products to be secure out-of-the-box. This in itself will significantly raise the game.

1.2 Value-Add: Why will a Vendor or Customer use this Self-Assessment Scheme?

Raise the Bar for Effective Security

Of course, no vendor will want to publish a Self-Assessment Scorecard which shows a very low score. We know the self-assessment searching questions are not trivial. We’re not expecting any vendors initially to score “Good” across the board. So, we anticipate the initial impact will be that security product vendors will keep their self-assessment results internal.

We do hope, however, that they will show that they are using the Self-Assessment Scheme and that it helps them to raise their game, and so raise the bar towards establishing a more secure marketplace where products are acceptably secure out-of-the-box. Product features are largely market-driven, so if vendors actually find they are losing market share because they score low on this (or any other credible) Self-Assessment Scheme, then that alone demonstrates its value.

Be Prepared with Good Answers

Since the self-assessment questions are available to both vendors and customers, we anticipate that vendors will want to make their own self-assessment of what their product looks like from a Jericho Forum security standpoint – how ready it is to operate effectively in a rapidly de-perimeterizing world? – so that they have ready answers to questions their customers are likely to raise.

Future-Proofing

After a while, we hope that most vendors will have used this Self-Assessment Scheme to improve their products and solutions sufficiently to score well in the self-assessment. How will customers then be able to differentiate the competing vendor products? There are two considerations here:

- If most (and hopefully all) vendors eventually achieve near-perfect scores on the self-assessment, then we will have succeeded in raising the security bar for the entire industry – which is a huge win for everyone.
- Of course we know that nothing is perfect, so we expect to revisit our set of searching questions in future, in the light of vendor and customer feedback and experience, and revise them to optimize how well they differentiate the key features and so deliver best value to both vendor and customer communities.

1.3 Self-Policing

As a Self-Assessment Scheme, neither the Jericho Forum nor its trademark holder The Open Group take any responsibility for validating self-assessment scores or associated claimant information. This scheme is entirely self-policing. It relies on the honesty of the submitters of Self-Assessment Scorecards, in the knowledge that their reputation will be tarnished if their Self-Assessment Scorecard is exposed as including false claims. This self-policing approach achieves the Jericho Forum goal of promoting industry awareness and use of our commandments while making the scheme very low-cost and low-maintenance.

1.4 Initial Industry Response

The response to-date has been encouraging.

A number of vendors have indicated that they welcome a tool like this because it enables them to differentiate their product from competing products that don't perform as well in practice. Also, because it is a Jericho Forum scheme, and the Jericho Forum is a totally independent thought-leader in this space, it provides the necessary degree of independence and objectivity that a vendor-driven scheme could not achieve.

Customer feedback similarly indicates a welcome for the scheme. Not surprisingly, as buyers they tend to be sceptical of vendor claims, so they welcome having an objective process that enables them to verify vendor claims on specified key features.

2 Using the Self-Assessment Scheme

2.1 Caveats

- This Jericho Forum Self-Assessment Scheme is not a formal certification process. It has no relationship or links with The Open Group certification programs.
- It is a deliberately lightweight management process intended to provide a framework for vendors of products and solutions to claim how well their product/solution satisfies the Jericho Forum commandments.
- The Self-Assessment Scheme is intended to be self-policing.
- This scheme must not be used to represent a measure of how secure an overall application or device is, nor should it be used to support any statement that purports to assert that any product or solution of any kind is “fit-for-purpose”.

2.2 How to Use this Document

As a vendor or customer:

- This document is open source and royalty-free. There is no cost impact to use it in any way you wish.

As a vendor of a product:

- You may use this document to complete the Self-Assessment Scorecard – see Section 4.
- Some commandments may not be applicable to your product, in which case the relevant part in your Self-Assessment Scorecard should be used to indicate this.
- You may use and publish your results in any form you wish, but you must not claim or imply that the Jericho Forum (or its trademark-holder, The Open Group) have in any way endorsed the results.

As a prospective purchaser of a product:

- As part of any purchasing decision, we recommend that you ask the vendor to provide a detailed self-assessment of their product in terms of their answers to questions on each of the 11 commandments.
- Should this not be available, then we recommend that you incorporate this document into any tender or Request for Quote (RFQ) that you issue to prospective vendors. If you wish to edit the questions or criteria in this Self-Assessment Scheme to better meet your requirements, then you are free to do so. If you do this, however, you may not then use

the term “Jericho Forum Self-Assessment” to refer to its use (except in a “derived from” acknowledgement) or to the resulting self-assessment score(s).

- If the vendor of the product offers you a self-assessment summary, then we recommend you compare their assessment against the complete Self-Assessment Scheme document to ensure their summary is complete, and evaluate how far you agree with their self-assessment results.

2.3 How to Complete a Self-Assessment

The template self-assessment table for each of the 11 commandments is presented in Section 3. Each self-assessment table includes:

- A statement of the commandment
- A set of questions to be answered on specific key aspects implicit in that commandment
- Columns for “Acceptable” and “Good”, in which the criteria for scoring one or the other against each question is described
- A note explaining how to arrive at an overall score for that commandment

Having completed answering all questions in all applicable self-assessment tables, and arrived at an overall score for each commandment:

- Enter these scores in the Self-Assessment Scorecard – see Section 4.
- Complete the identification details on the Self-Assessment Scorecard.
- Use the Scorecard and the detailed Self-Assessment in whichever way you find beneficial, subject to the constraints expressed earlier in this section.

3 Self-Assessment Template

The following section presents each of the 11 Jericho Forum commandments in turn, along with a self-assessment table listing the key requirements implicit in that commandment, and the criteria for scoring “Acceptable” or “Good” against each requirement.

Guidance notes preceding each self-assessment table explain how to arrive at an overall score for that commandment. For any given product, answer each requirement, assessing it as “Acceptable” or “Good”. If it achieves neither, then the score is “Unacceptable”. As already explained, the objective is to show where and how to improve the security capabilities of products, and initially we are not expecting vendors to score all “Good” on each commandment.

Then, enter the score for each commandment into the Self-Assessment Scorecard (see Section 4) to arrive at a summary Self-Assessment Scorecard.

If a commandment does not apply to your product, then indicate this in the Not Applicable column of the Scorecard for that commandment.

We encourage you to use your Self-Assessment in whichever way you find beneficial, subject to the constraints expressed in Section 2 of this Guide.

Any queries may be addressed to jerichoforum-interest@opengroup.org.

1. The scope and level of protection should be specific and appropriate to the asset at risk.

Introduction

Risk should always be considered in the context of where the application/service/solution will be used. Thus, the ultimate assessment of risk will be in the domain of the purchaser of the product.

However, the vendor also plays a major part to ensure that all the facts are available to prospective purchasers, thus enabling a proper risk assessment to be made.

Vendors will also see where their solutions are used and therefore should understand where successful implementations are implemented, and more importantly where other clients have encountered problems.

Typically, a vendor should define the limits and/or risks in using their product or solution in particular environments. This may be regulatory (or lack of compliance or approval with a particular regulatory environment) or may be due to a particular design constraint with the product.

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls in this section.

Compliance with particular regulations and standards should be stated.

Limits or design constraints should be identified here, particularly as a result of meeting a local regulatory environment; for example, *downgrading the encryption from AES because of a local requirement to use a particular approved encryption algorithm.*

Areas for which the product is not suitable should be clearly stated, such as *cannot be used in China as the encryption algorithms are not approved.*

Where assumptions about security or other controls are made they must be stated; for example, *this server must be operated in an access controlled server room, as admin access could be gained through physical access to the server.*

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #1	Risk	
	Acceptable	Good [Best Practice]
The scope and level of protection should be specific and appropriate to the asset at risk.	Businesses must understand the risk to the asset being protected prior to purchase and vendors should provide adequate documentation of the security model in use to enable the prospective vendor to perform a risk analysis and determine whether the proposed solution is fit-for-purpose.	To enable businesses to make a risk analysis, full disclosure is provided on types of use that the product is suitable for and clearly details areas where this product may have issues (legal, technical, etc.) that will need further investigation, or areas for which this product is not suitable.
Business demands that security enables business agility and is cost-effective.	Aspirational statement, not applicable to complete.	Aspirational statement, not applicable to complete.
Whereas boundary firewalls may continue to provide basic network protection, individual systems and data will need to be capable of protecting themselves.	The system or application should not be reliant on any external devices to provide an adequate level of security. There should be no constraint (topologically) where this device can be used.	The model used to protect and/or harden the product and/or service should be fully documented. Details of any assumptions made must be fully disclosed.
In general, it's easier to protect an asset the closer protection is provided.	A description of the protection method(s) used is provided. Clear explanation of how overall protection/security is achieved is provided.	The explanation of the security model should be detailed by Confidentiality, Integrity, and Availability (CIA) and ideally reference to the seven-layer stack and, where appropriate, physical controls.

2. Security mechanisms must be pervasive, simple, scalable, and easy to manage.

Introduction

As a rule of thumb, security solutions that are simple tend not only to work but also be implementable. By contrast, complex solutions that are not understood by the majority of people are bypassed, or are so complex they have loopholes in them.

Vendors should be able to provide a simple conceptual diagram describing the security model in use, where the protection is provided, and how the solution is managed, thus enabling understanding of how a secure solution is achieved.

Management of the solution should also be described – from ensuring that systems are updated (and describing how), to explaining how administrator access is gained (bearing in mind much maintenance will be performed remotely – refer to Jericho Forum Commandment #3), and how user accounts are managed – bearing in mind that data and objects should be managed consistently in one place – so not inventing its own user repository, but instead the default should be to leverage the existing repository of users (e.g., Active Directory).

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls, further guidance, or examples in this section.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment # 2	Scalability	
	Acceptable	Good [Best Practice]
Security mechanisms must be pervasive, simple, scalable, and easy to manage.	Clear limits on size, scalability, and any defined limits on size (scalability). This should cover items such as number of users, number of servers, distributed servers, transaction limits, throughput limits, physical distances, etc.	For every architectural parameter of the system, their constraints (and potentially interdependencies) are clearly defined.
Unnecessary complexity is a threat to good security.	There is broad alignment to the task, with appropriate mitigation of risk. A simple security diagram is provided showing how security is architected into the solution to provide a secure solution.	The solution is singularly focused on the task at hand. Controls are singularly aligned to the task/mitigation of the risk.
Coherent security principles are required which span all tiers of the architecture.	There are acceptable levels of system integration with appropriate APIs provided. All APIs or other interfaces are fully documented to allow integration.	The application/system integrates into a centralized control system, allowing logging, etc. to take place.
Security mechanisms must scale; from small objects to large objects.	The management of security should demonstrably decrease as you get larger.	The system is capable of supporting small (10) to large (100,000+) objects/users. Expansion of the system should not detract from being able to perform granular management. Objects should be capable of being nested, such as users managed as groups, groups within groups, nested rules, etc.
To be both simple and scalable, interoperable security “building blocks” need to be capable of being combined to provide the required security mechanisms.	There is a full (and rich) definition of secure APIs or other secure interface standards (the majority of which use open standards) allowing centralized management/logging/integration/APIs. The glueing has been done for you.	All interfaces are secure and use open interface standards. Where propriety standards or APIs are in use, they are fully documented and free of all charges.

3. Assume context at your peril.

Introduction

Solutions are often designed with a particular solution, or context in mind. It is not uncommon to find solutions being designed for a particular client or in ignorance of what will happen when the solution is scaled to a global 24x7 environment.

Thus, a solution that requires a 4-hour downtime to close the database and backup may fail when used in a follow-the-sun model.

The aim here is to document the scope and limitations so that the end-user or prospective purchaser can make an informed choice. However, when an organization is new to this area, often the problem is with little prior experience they have not encountered the potential problems. Here the vendor bears a responsibility to clearly explain limitations both in design and found in practice to “bootstrap” a prospective purchaser’s knowledge.

Note: While the normal “sales” approach is often to hide the limitations of a product, vendors can often overcome this by requesting competing bidders to provide their competitive intelligence on the shortlisted products and solutions.

Compensating Controls, Guidance, and Examples

The compensating control should clearly state what design assumptions were made when the product was architected. Look at drug packaging for good examples where detailed instructions are included in the packaging explaining how products are only to be used in certain conditions, and side-effects are clearly described.

Examples in the computer industry are that a product may only support a number of users, or authenticate against Active Directory Usernames, but not Active Directory Groups.

Other examples may be that the encryption used is of a particular type and thus not legal to be used in certain countries.

Most of this item is about good, open, and honest documentation, freely available, documenting the limits of a solution/device/product.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #3	Context	
	Acceptable	Good [Best Practice]
Assume context at your peril.	Do not assume your organization has full control over the data, environment, and processing. Where assumptions have been made, these must be fully documented.	The limitations of the solution are clearly documented and available for free unrestricted download. Documentation of such limits is in plain language (not assuming the context of a technically literate reader).
Security solutions designed for one environment may not be transferable to work in another. Thus, it is important to understand the limitations of any security solution.	The application of the solution/model should be documented with assumptions and limitations stated. The documentation should clearly state the environment this is designed to operate in and when it is inappropriate to use the solution. A good (secure) practice guide should be provided.	The solution is designed to scale to a large distributed environment and operate securely on the raw Internet. Documentation clearly states how to scale the solution from small implementation to large corporate solution. Documentation clearly states the data model used and any assumptions when protecting that data. The documentation clearly states context in which the system and hardware is designed to be used, the limitations of those assumptions, and methods for augmenting those controls (if required).
Problems, limitations, and issues can come from a variety of sources, including geographic, legal, technical, acceptability of risk, etc.	The application of the solution/model should be documented; any issues from geographic, legal, technical, acceptability of risk, etc.	The solution is capable of identifying and adapting to changing or different contexts.

4. Devices and applications must communicate using open, secure protocols.

Introduction

Any application or device will communicate using a set of protocols. It is therefore essential that those protocols are known, fully documented, and appropriate. In addition, these protocols should ideally be open and secure, and should either be the only option or the default option.

Compensating Controls, Guidance, and Examples

Where the purpose of a device or application is to deliver public information to anyone who wants it, then clearly, using http (web access) or TFTP (unauthenticated file transfer) is appropriate, warranting an “Acceptable” rating. One could argue that https would be better as it gives a degree of certainty that you are connected to and downloading information from the correct site.

Outside of that public interface, all administrative interfaces must use inherently secure protocols, and all passing of non-public information must use inherently secure protocols.

Where files are being transferred, it may be determined that the data will be protected at file level (say through encryption with a pre-shared key) and then the protocol used could be TFTP – this would rate “Acceptable”. Ideally the two systems should identify whether they are permitted to talk to each other (using say PKI keys, but not IP or MAC addresses) and then use TFTP to exchange encrypted files – this would warrant a “Good”.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.
4. Where an application or device-specific aim is to deliver unclassified/public information to anyone who wishes to access it, then an insecure (open) protocol may be justifiable. Examples would be a brochure-ware web server.
5. An example of a specific (and appropriate) protocol is the GSM A5/3 voice protocol, which provides secure voice communication but is designed to maintain time integrity. Compare against voice over IPSec (designed to maintain data integrity) and thus is an inappropriate protocol. Compare against voice over IPsec, designed to maintain data integrity but not time integrity, so is an inappropriate protocol.
6. Documentation should typically be provided with the software and/or manual. Providing documentation on-demand or via a support web site where the end-user would need to search out would not be acceptable to score a “Good”.

Jericho Forum Commandment #4	Protocols			
	Application		Device	
	Acceptable	Good [Best Practice]	Acceptable	Good [Best Practice]
Devices and applications must communicate using open, secure protocols.	The application by default only uses inherently secure protocols. ⁴	All protocols in use are documented. ⁶ Where there is an option to use an insecure protocol, then there are clear, documented guidelines on the issues and additional security controls that should be implemented.	The device by default uses only inherently secure protocols; ⁴ by default all other ports are disabled.	All protocols in use are documented. ⁶ Where there is an option to enable an insecure protocol, then there are clear documented guidelines on the issues and the additional security controls that should be implemented.
Security through obscurity is a flawed assumption – secure protocols demand open peer review to provide robust assessment and thus wide acceptance and use.	All protocols in use conform to open standards and all encryption used is open and has been subject to peer/industry review.	There are no cost implications (licensing, royalty, or other) from using any of the protocols.	All protocols in use conform to open standards and all encryption used is open and has been subject to peer/industry review.	There are no cost implications (licensing, royalty, or other) from using any of the protocols.
The security requirements of confidentiality, integrity, and availability (reliability) should be assessed and built in to protocols as appropriate, not added on.	The protocol(s) used are appropriate for the task(s) being undertaken by the application.	The inherently secure protocol(s) are specific ⁵ to the task being undertaken by the application.	The protocol(s) used are appropriate for the task(s) being undertaken by the application.	The inherently secure protocol(s) are specific ⁵ to the task being undertaken by the application.
Encrypted encapsulation should only be used when appropriate and does not solve everything.	The application can function, by default, with only inherently secure protocols.	No insecure protocols are in use and encapsulation is not required.	Any insecure protocol is encapsulated in an inherently secure protocol.	No insecure protocols are in use and encapsulation is not required.

5. All devices must be capable of maintaining their security policy on an un-trusted network.

Introduction

When a device is connected to a network, no assumption should be made by the application/device vendor about the state of the network. The working assumption should be that the network is un-trusted. It is imperative that the vendor understands, designs, and then explains how their solution works in normal use, is managed (super-user access), updated, and interacts with other typical components.

Compensating Controls, Guidance, and Examples

Where security of the device/application is unable to be assured in its native state, then the vendor should specify exactly what additional components should be added to deliver a holistic and secure solution. This should be documented, with examples, clearly demonstrating how secure operation of all aspects of operational usage is achieved.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #5	Device	
	Acceptable	Good [Best Practice]
All devices must be capable of maintaining their security policy on an un-trusted network.	<p>The device(s)/systems security model is designed to work identically on an un-trusted (hostile) network, and any Intranet.</p> <p>All device services, applications, etc. must support this.</p>	<p>All devices are architected; non-de-perimeterized issues are mitigated with appropriate solutions.</p> <p>The architecture is designed to operate in an un-trusted environment.</p> <p>Backwards compatibility to insecure, or non-de-perimeterized solutions is not permitted.</p> <p>All device services are externalized and allow identical management on or off-Intranet.</p>
A “security policy” defines the rules with regard to the protection of the asset.	<p>The security policy requires no reference to the connection parameters and/or the location.</p> <p>A minimum level of security state can be maintained and that minimum state updated regardless of environment or location.</p> <p>The solution will fail safe should an unacceptable level of threat be encountered.</p>	<p>The state can be maintained and/or updated dependent on the risk.</p> <p>The device/systems can audit and report on the configuration, current state, current risk, and current threats, with ability to be able to report and confirm regardless of environment or location.</p> <p>The device/systems can self-heal, repair itself, and be fully managed.</p>
Rules must be complete with respect to arbitrary contexts.	<p>Security solutions require no reference to the connection parameters/location.</p> <p>There are no caveats or footnotes to the rule set.</p>	<p>A risk and controls review/analysis has been carried out and is freely available.</p> <p>The risk and controls review has addressed all possible (feasible) operating contexts, is documented, and freely available.</p>
Any implementation must be capable of surviving on the raw Internet; e.g., will not break on any input.	<p>The device(s) security model is designed to work identically on an un-trusted (hostile) network, and any Intranet.</p> <p>The device must not accept any management packets that are not specifically meant for the device.</p>	<p>All inputs for the device will be rejected if not certified for the device.</p> <p>All data is validated and rejected if inappropriate.</p>

6. All people, processes, and technology must have declared and transparent levels of trust for any transaction to take place.

Introduction

Trust needs to happen at all levels of a transaction, whether it is user to user, user to machine, or machine to machine.

For a transaction to take place it is essential that systems can communicate trust levels and the attributes associated with those trust levels.

For a vendor, they should document the trust model being used, and also those attributes that go to make up that trust, plus any assumptions they make about devices and/or people and the attributes they can provide.

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls, further guidance, or examples in this section.

Notes

1. "Good" is a further build on "Acceptable" – to achieve a "Good", the criteria for "Acceptable" must be met as well.
2. (Scoring) To achieve an overall "Good", a rating of "Good" in all areas must be achieved.
3. (Scoring) To achieve an overall rating of "Acceptable", a rating of "Good" and/or "Acceptable" in all areas must be achieved.
4. "Contract" includes agreed understandings between collaborating parties, not just legal contracts. They include electronically brokered agreements, as defined in the Jericho Forum position paper on "Collaboration Oriented Architectures".

Jericho Forum Commandment #6	Trust	
	Acceptable	Good [Best Practice]
All people, processes, and technology must have declared and transparent levels of trust for any transaction to take place.	The trust model in use is fully documented for all transactions, transaction routes, inputs, and outputs.	Access (device, systems, application) is restricted / modified based on the strength of the trust model.
Trust in this context is establishing understanding between contracting parties to conduct a transaction and the obligations this assigns on each party involved.	For all transactions, a contract (see Note 4) is in place with clearly understood obligations and rights on all sides.	Each transaction can be traced to the applicable contracts (see Note 4).
Trust models must encompass people/organizations and devices/infrastructure.	The documentation of the trust model covered how all aspects of trust is achieved [people, organizational, devices, infrastructure].	The trust model is dynamic based on changing environment, circumstances, or changing reputation.
Trust level may vary by location, transaction type, user role, and transactional risk.	The trust model in use can adapt to changing circumstances such as location, type (value) of transaction, and risk associated with the transaction.	Both parties in the transaction are considered in the adapting circumstances.

7. Mutual trust assurance levels must be determinable.

Introduction

Trust needs to be bi-directional. For example, financial banks need to be sure it's actually their customer who is connecting, and their banking customers need to ensure it's actually their bank and not a phishing site they are transacting with.

Devices need to ensure that the latest firmware is from their vendor and not a rogue update with a back-door added.

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls, further guidance, or examples in this section.

Notes

1. "Good" is a further build on "Acceptable" – to achieve a "Good", the criteria for "Acceptable" must be met as well.
2. (Scoring) To achieve an overall "Good", a rating of "Good" in all areas must be achieved.
3. (Scoring) To achieve an overall rating of "Acceptable", a rating of "Good" and/or "Acceptable" in all areas must be achieved.

Jericho Forum Commandment #7	Trust			
	Devices		Users	
	Acceptable	Good [Best Practice]	Acceptable	Good [Best Practice]
Mutual trust assurance levels must be determinable.				
Devices and users must be capable of appropriate levels of (mutual) authentication for accessing systems and data.	End-points can mutually authenticate each other.	End-points can mutually authenticate each other according to externalized policies.	Parties can mutually authenticate each other.	Parties can mutually authenticate each other according to externalized policies.
Authentication and authorization frameworks must support the trust model.	Access policies can be related to relevant risk factors.	Access policies are externalized from the end-point.	Access policies can be related to relevant risk factors.	Access policies are externalized from the end-point.

8. Authentication, authorization, and accountability must interoperate/exchange outside of your locus/area of control.

Introduction

Often vendors provide solutions with an in-built solution for user management, authorization, and accountability.

Such management of a localized and highly defined/controlled ecosystem, while useful from a vendor point of view, is rarely ideal in the real world, where the aim is to manage a user once and consistently.

As organizations operate outside their own boundaries, so the ability to federate that management becomes essential; otherwise, organizations end up managing users for whom they do not manage the primary information source related to that user.

This also applies to machine authentication.

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls, further guidance, or examples in this section.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #8	Control	
	Acceptable	Good [Best Practice]
Authentication, authorization, and accountability must interoperate/exchange outside of your locus/area of control.	Authentication: The ability to use and validate the identity of external entities (people and/or systems and/or devices).	Ability to use and validate the identity of entities independent of organization or location. Directly consume the external identity from multiple sources, to open standards.
	Authorization: The ability to link the authentication to the authorization system.	Consuming claims, link claims to authorization. Ability to link who, what (requested), when, how, where, to a risk-based access model.
	Accountability: The ability to determine (audit) who, what, when, how, where, for internal access.	Ability to determine who, what, when, how, where, independent of organization or location.
People/systems must be able to manage permissions of resources and rights of users they don't control.	Access to YOUR data/system can be controlled by your ACLs/rules including users you don't control. Granular user/system access control to resources.	Claims/context rules-based access control to resources based on multiple applicable factors.
There must be capability of trusting an organization, which can authenticate individuals or groups, thus eliminating the need to create separate identities.	Capability exists to accept user credentials from third parties (either through company or individual). Ability to on-board and off-board organization, and/or trust brokers (need business process behind this).	Ability to on-board and off-board individuals. Visibility into what is being professed from the third-party trust organization – transient trust from validation provider. Support of delegated authorization from organizations and individuals.
In principle, only one instance of person/system/identity may exist, but privacy necessitates the support for multiple instances, or one instance with multiple facets.	Ability to authenticate an identity against a range of repositories.	Single identity and able to manage multiple claims against it.
Systems must be able to pass on security credentials/assertions.	Transitive claims are supported.	Bi-directional validation required; respond to and request security credential and assertions. Pass-on (transitive) claims.
Multiple loci (areas) of control must be supported.	Access to systems and/or data may be achieved using multiple sources of authentication and credentials.	Handle combination of claims with multiple attributes, and modify claims dependent on other attributes.

9. Access to data should be controlled by security attributes of the data itself.

Introduction

Data, and access to that data, is (generally) what we are trying to secure. It is therefore essential that the data model for securing that data is fully understood.

Usually a data flow diagram will aid understanding of a system, from user access, interaction with other programs (maybe through APIs), and then storage, both on systems and on backups.

Data in all of its states should have its access controlled appropriately. This is especially necessary at the points where data transits outside of the system under consideration and into foreign systems, or worst-case is roaming freely on the Internet.

Ideally where the data model means that data will transit outside the system, then consideration must be given to how that data will be protected. Where access was controlled within the systems, then such access constraints should continue, either by natively protecting the data itself, or by transferring that access information along with the data (where you trust the receiving system). When transferring data, attributes may be negotiated to meet the receiving system's policies, with the agreement of the sending system.

Such solutions could involve Digital Rights Management (DRM) applied to the data itself; or transfer to a foreign system for which a trust relationship exists; or transfer to a foreign device where a cryptographically secure container has been established under your control to hold your data.

Compensating Controls, Guidance, and Examples

There should not be any need for compensating controls, further guidance, or examples in this section.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #9	Access to Data	
	Acceptable	Good [Best Practice]
Access to data should be controlled by security attributes of the data itself.	The solution should be able to support a wide range of data classifications from public to confidential.	Data should be linked to (or have embedded) a rich set of metadata which will control access based on who, where, when, what, etc. When transferring data to other parties, attributes of the data may be negotiated.
Attributes can be held within the data (DRM/metadata) or could be a separate system.	Data and metadata is irrevocably bound together.	By default, attributes should be inherited when copied. An attribute should be the ability to modify the attributes!
Access/security could be implemented by encryption.	Where the data is required (by risk assessment/classification) to be confidential, then strong encryption should be used to protect it.	Encryption key management is open, flexible, and scalable and temporal.
Some data may have “public, non-confidential” attributes.	Where the data needs high integrity, then a cryptographic fingerprint is used.	There is support for mixed data, and behaviors of that mixed data.
Access and access rights have a temporal component.	The ability exists to define changes in attribute based on time (for example, an embargoed press release).	The attributes have the capability to void old or expired data.

10. Data privacy (and security of any asset of sufficiently high value) requires a segregation of duties/privileges.

Introduction

Applications or devices that handle any kind of sensitive data – personal, intellectual property, financial – should be capable of providing the appropriate level of segregation. This segregation should ideally be capable of more than just binary control (Joe is a user, John is an admin, or Jim does a particular role). In a de-perimeterized world, where the levels of trust are key to decision-making, the device and application should be able to utilize this trust information to allow/modify the access being granted.

Compensating Controls, Guidance, and Examples

Only where the device/application can have no possibility of holding assets of high value can an application or device be exempt from this commandment. Thus, a Blu-ray player or TV that has an ethernet connection to update its firmware may be exempt (but not from the need to ensure the code update is trusted); but a database cannot.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #10	Access to Data	
	Acceptable	Good [Best Practice]
Data privacy (and security of any asset of sufficiently high value) requires a segregation of duties/privileges.	All data is assessed for confidentiality. Access to data is managed against individual user rights (potentially role-based).	Data is also assessed for integrity and availability.
Permissions, keys, privileges, etc. must ultimately fall under independent control, or there will always be a weakest link at the top of the chain of trust.	Key management and key escrow is under a separate chain of access to the data with no overlap of personnel or administrators.	Data is regularly monitored for segregation of duties with inappropriate access levels highlighted. The need for continuing access is regularly reviewed and logged.
Administrator access must also be subject to these controls.	Administrators for key management should not overlap with data administrators.	Administrators for key management should be managed via an independent authentication system with a higher level of validation and logging. Logging should be to WORM (or similar) media and should be tamper-evident.

11. By default, data must be appropriately secured when stored, in transit, and in use.

Introduction

Data needs to be secure for a number of reasons – from providing the integrity of that data, to ensuring that it is properly protected from being viewed by unauthorized persons or systems.

The use of a risk analysis should be undertaken to understand the potential risks the data may be subject to, and then appropriate measures implemented.

Providers of devices and systems should understand the worst-case requirements their product will be used for and provide.

Compensating Controls, Guidance, and Examples

Data intended to be completely in the public domain – i.e., totally available to all, with no requirements on confidentiality or integrity – requires no measures to secure that data. Otherwise, compliance with this requirement applies.

Notes

1. “Good” is a further build on “Acceptable” – to achieve a “Good”, the criteria for “Acceptable” must be met as well.
2. (Scoring) To achieve an overall “Good”, a rating of “Good” in all areas must be achieved.
3. (Scoring) To achieve an overall rating of “Acceptable”, a rating of “Good” and/or “Acceptable” in all areas must be achieved.

Jericho Forum Commandment #11	Access to Data	
	Acceptable	Good [Best Practice]
By default, data must be appropriately secured when stored, in transit, and in use.	<p>All data is assessed for confidentiality.</p> <p>All confidential data is adequately protected against theft (physical & logical).</p> <p>Data in transit uses only inherently secure protocols (see Jericho Forum Commandment #4).</p> <p>Data in-use only operates in a secured environment.</p>	<p>Data is also assessed for integrity. All confidential data is encrypted when stored. Key escrow is fully managed to ensure long-term recovery. The integrity of the “in-use” environment can be validated, and repatriation/destruction of the data guaranteed.</p>
Removing the (secure) default must be a conscious act.	<p>There is no alternative other than “secure”.</p> <p>The user must be made fully aware of the consequences of their actions.</p>	<p>Security enforced by security policy, or the data itself, with removal logged, or removal not possible.</p> <p>Removal by translation of the data is also enforced.</p>
High security should not be enforced for everything; “appropriate” implies varying levels with potentially some data not secured at all.	<p>Data is protected appropriate to the highest level of classification of the data being handled.</p>	<p>Systems should use the data’s classification to appropriately apply variable levels of protection appropriate to the risk the data will be subjected to.</p>


4 Self-Assessment Scorecard

Having completed answering all the self-assessment questions for all the applicable commandments, and arrived at an overall score for each commandment:

- Enter these scores in the Self-Assessment Scorecard – see below.
- Complete the identification details at the head of the Self-Assessment Scorecard.
- Add any relevant Notes or Observations you wish to make to clarify your answer to any specific questions in any commandment.
- Then use the Scorecard and the detailed Self-Assessment in whichever way you find beneficial, subject to the constraints expressed earlier in this section. In particular, we hope you will review the areas where your score is not “Good”, with a view to taking appropriate measures to improve the security of your product such that it achieves “Good”.

Jericho Forum® Self-Assessment Scorecard

Product Name: _____ **Vendor Name:** _____
Product Type(s): _____
Description: _____
Version Number: _____ **Date (dd/mm/yyyy):** _____
Named Controllers: _____
Assessor Name: _____ **Assessor Organization:** _____

		Not Applicable (enter "X")	Not Acceptable (enter "X")	Acceptable (enter "X")	Good (enter "X")	Notes/Observations
1	Specific & appropriate to the asset at risk					
2	Security, simple, scalable, & manageable					
3	Assume context at your peril					
4	Open & secure protocols					
5	Maintain security policy on un-trusted network					
6	Transparent trust					
7	Mutual trust assurance levels					
8	Authentication outside of locus of control					
9	Access by security attributes of the data					
10	Data privacy requires segregation of duties					
11	Data appropriately secured					
Overall self-assessment of software or device:						

Additional Notes and Observations

Commandment	Notes/Observations
1	<none>
2	<none>
3	<none>
4	<none>
5	<none>
6	<none>
7	<none>
8	<none>
9	<none>
10	<none>
11	<none>