

Manoranjana Paul



!= Marijuana Paul



Entertainment Paul





Entertainment + Education ==
Enlightenment



Entertainment - Education ==



Main Entry: **joker**  [joh-ker]  [Show IPA](#)

Part of Speech: *noun*

Definition: person who kids, teases

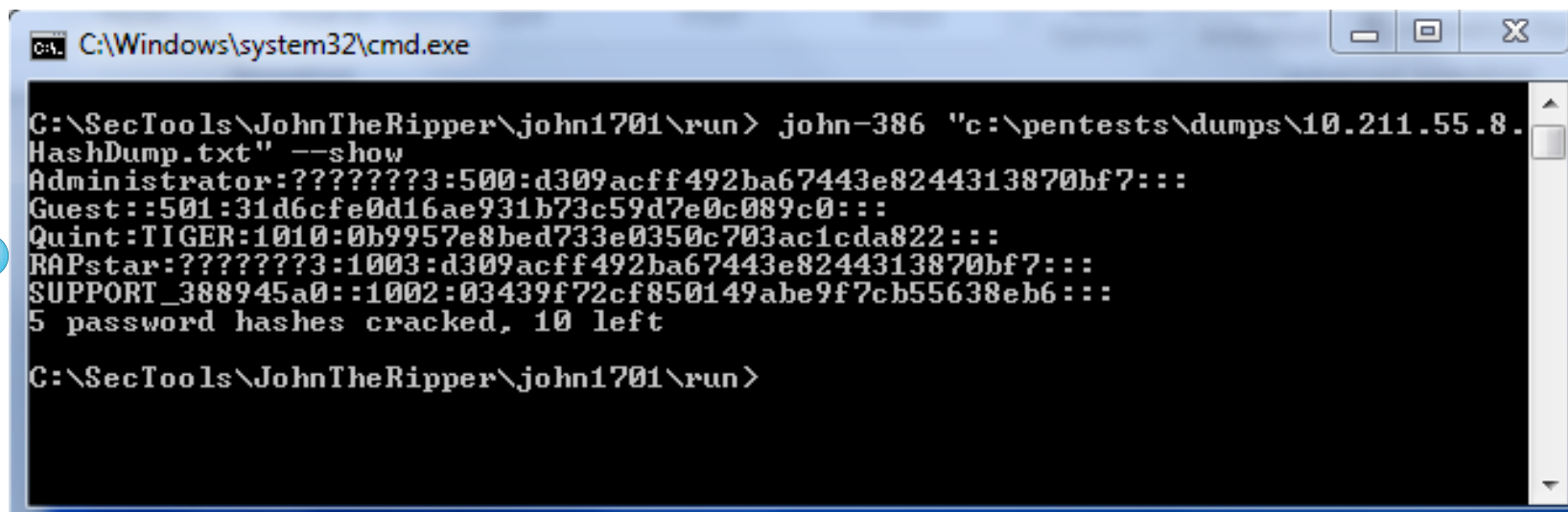
Synonyms: actor, banana, buffoon, card*, clown, comedian, comic, cutup, droll, farceur, fool, funster, gagster, humorist, jester, jokesmith, jokester, joshier, kidder, life of the party, prankster, punster, quipster, second banana, stand-up comic, stooge, straight person, top banana, trickster, wag, **wisecracker**, wit

* = informal/non-formal usage



wise





```
C:\Windows\system32\cmd.exe

C:\SecTools\JohnTheRipper\john1701\run> john-386 "c:\pentests\dumps\10.211.55.8.
HashDump.txt" --show
Administrator:???????3:500:d309acff492ba67443e8244313870bf7:::
Guest::501:31d6cfe0d16ae931b73c59d7e0c089c0:::
Quint:TIGER:1010:0b9957e8bed733e0350c703ac1cda822:::
RAPstar:???????3:1003:d309acff492ba67443e8244313870bf7:::
SUPPORT_388945a0::1002:03439f72cf850149abe9f7cb55638eb6:::
5 password hashes cracked, 10 left

C:\SecTools\JohnTheRipper\john1701\run>
```

cracker





wise





is wise





Christian



“L33t”



“L4m3”





After 2 near death calls

Christian





www.hackformers.org

@hackformers



Teach Security



Teach Christ

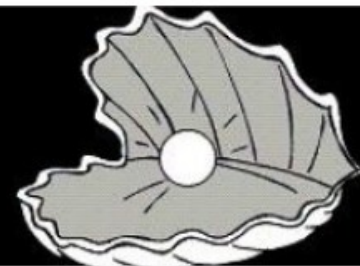


Teach Security in Christ



Hidden Treasures

To get pearls, one MUST dive deep



<http://www.facebook.com/getpearls>

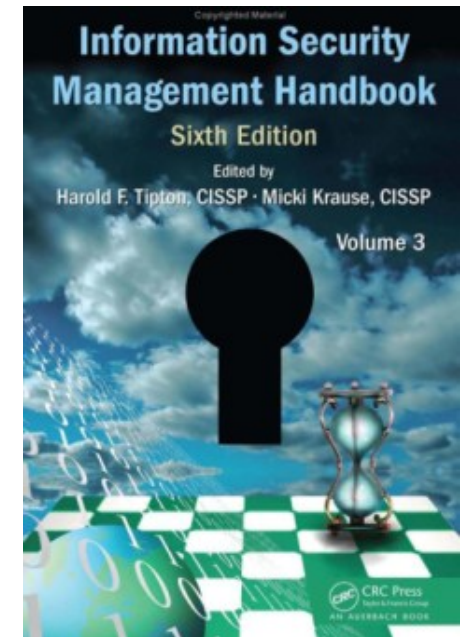
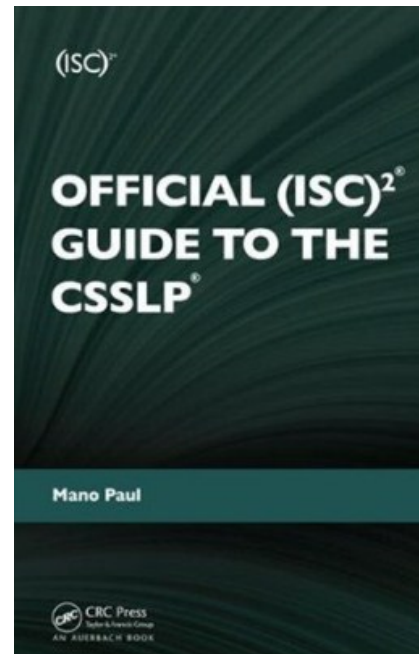
<http://thepauls.wordpress.com>





Author







Advisor

Software Assurance





(ISC)²®





And a few more

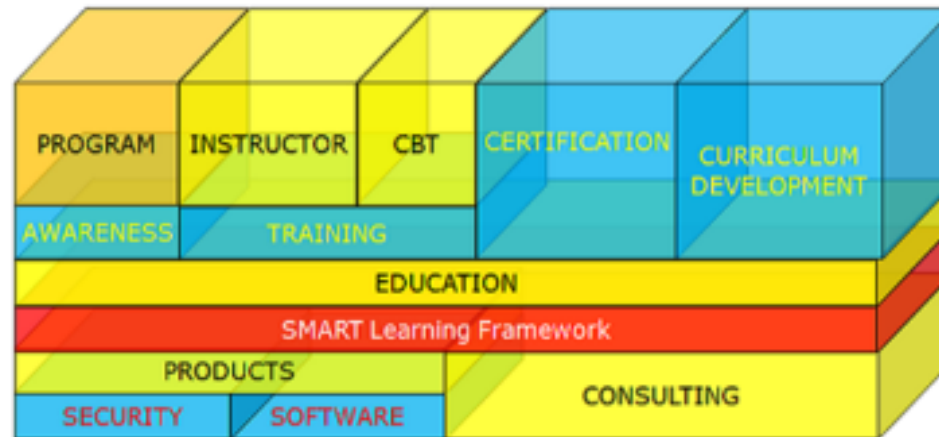
- MCAD
- MCSD
- ECSA
- CompTIA Network +





SecuRisk Solutions

SecuRisk Solutions Framework



*SMART - Skills Measuring Assessment Reinforced Training

Training
Products
Consulting



Express Certifications

Certification Practice Tests

CISSP

CSSLP

SSCP

CAP









```
1 <?php
2 // Run CSRF check, on
3 NoCSRF::check( 'csrf'
4
5 // Generate CSRF tok
6 $token = NoCSRF::gen
7
8
9 ?>
```



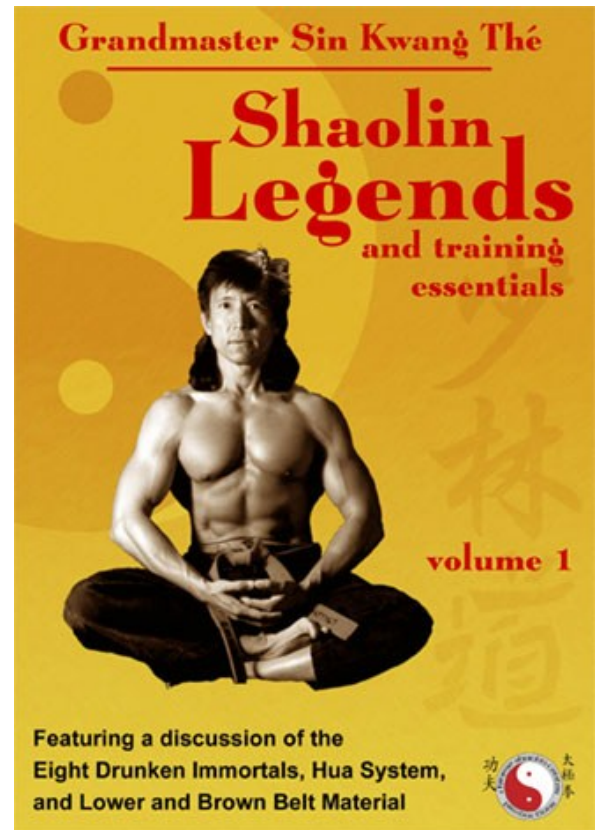






‘dash4rk’









2nd Degree Brown Belt




Black Belt









My son once asked me
“Dada, Are I Famous?”







PLAYBOY

STEPHEN KING, TOM MCGUANE,
WILLIAM F. BUCKLEY, JR.,
LEONARD MICHAELS, LARRY
L. KING, D. KEITH MANO,
PAUL ERDMAN, G. GORDON
LIDDY, EDDIE MURPHY,

**NOT
ME**




LIAM F. BUCKLEY, JR.,
ED MICHAELS, LARRY
KING, D. KEITH MANO,
ERDMAN, G. GORDON
LIDDY, EDDIE MURPHY,
HERSCHEL WALKER, DAN
GREENBURG AND
SUZANNE O'MALLEY
PLUS: SHANNON TWEED
LOVINGLY
PHOTOGRAPHED BY
GEORGE HURRELL,
CARS '83 AND A REVIEW OF THE
PAST YEAR'S DELIGHTFUL DOZEN PLAYMATES



So who am I?






Christian

Author-Biologist-CEO-Dash4rk

ABCD





Love my Savior,
Love my Spouse,
Love my Sons,
Love Shaolin,
Love Sharks,
Love Security





Mano 'dash4rk' Paul





The 7 Qualities of Highly Secure Software



Disclaimer

- !Pimp my book talk
 - One time on a flight ... someone asked me
 - What is this book about?
 - Is it any good?
- All opinions expressed are my own and not reflective of my employerWait a minute!
- Tweet/Facebook/Blogs ... permission?

What we ...

- Produce
 - Insecure (Hackable) Software
- Need
 - Highly Secure Software

What is this talk about?

- Not about
 - 7 things I need to put in my code (software)
- About
 - 7 things you should take into account when
 - Designing
 - Developing
 - Deploying Software.
- Technical – Operations – Management focused

7 Myths to bust

- #1 – We have a firewall
- #2 – We use SSL
- #3 – We have IDS/IPS
- #4 – We are not be accessible from the Internet
- #5 – We have never been compromised
- #6 – Security is “Not my job”
- #7 – Security adds little/no business value

What is Highly Secure Software?

- Hacker-proof
- 3Rs of Software Assurance (Trust)
 - Reliable
 - Resilient
 - Recoverable

007 ...

- #1 – Security is Built In, Not Bolted On
- #2 – Functionality Maps to a Security Plan
- #3 – Includes Foundational Assurance Elements
- #4 – Is Balanced
- #5 – Incorporates Security Requirements
- #6 – Is Developed Collaboratively
- #7 – Is Adaptable

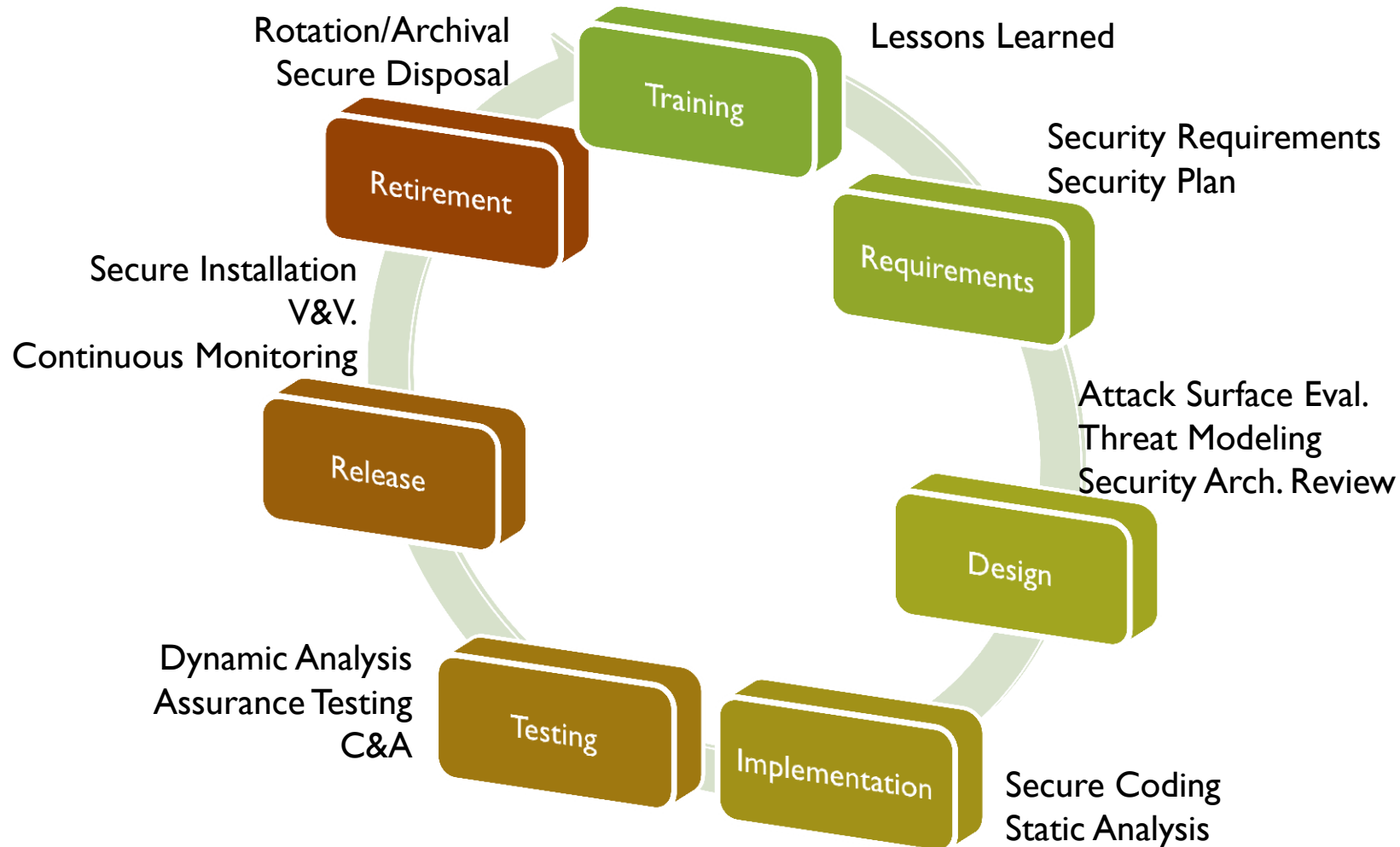




#1 – Security is Built In, Not Bolted On

- The Ant and the Grasshopper
- Be proactive not reactive
- Be strategic and not just tactical (Tool centric)

Security Development Lifecycle



Building Security In

- MOM in Cybercrime
 - Motive ? Hacker Motivations
 - Opportunities < Reduced Attack Surface
 - Means < Controls to Mitigate
- Security Processes and Implementing Controls
- Integrated with the SDLC
 - Requirements to Release ... is there more?



#2 – Functionality Maps to a Security Plan

- Breaking the Tape
- Begin with the End in Mind
 - How “secure” is your software going to be?
- Functionality \leftrightarrow Controls in Security Plan

Security Plan

- Framework for 'Assurance' Foundation
- Failing to plan = planning to Fail
- Overview of *applicable* security requirements
 - External (GRC+P)
 - Internal (Policies/Standards)
- Controls
 - Safeguards / Countermeasures
 - Technical (System) / Operational (People) / Management (Risk based)

Mapped Software

- **Functionality:** Each user must have an unique account for interacting with the software.
- **Controls:** Unique usernames and passwords
- **Security Requirements:** Remove test and default accounts before release (PCI DSS 6.3.1)
- **Threat:** Impersonation and Repudiation



#3 – Includes Foundational Assurance Elements

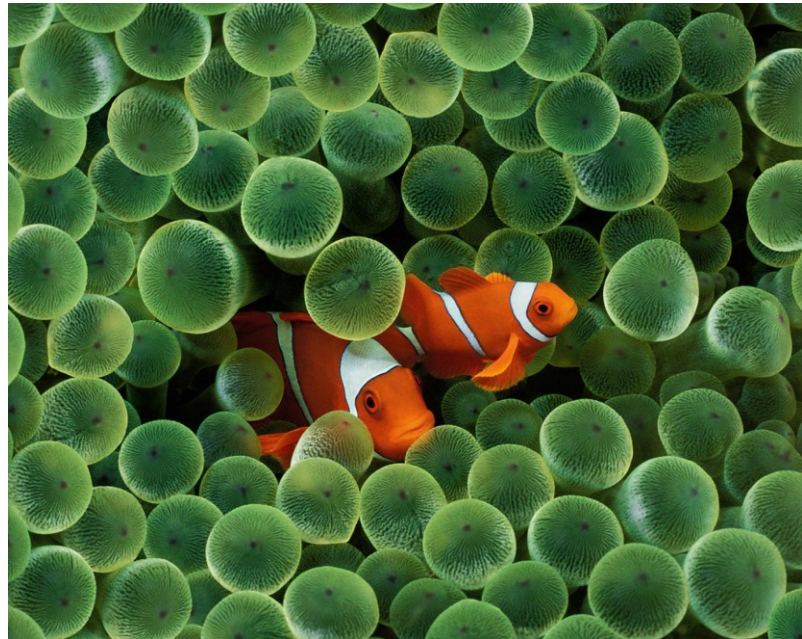
- What lies beneath?
- Put first things first

First things First



#4 – Is Balanced

- The Clown Fish and the Anemone
- Think Win/Win



Balancing what?

- Risk and Reward
 - Security Lingo (ROI)
- Functionality and Assurance
 - Iron Triangle Triple Constraints
 - *“It is a real trade off. You always want the functionality and you always know you are providing opportunities so you need to take that into account and try to build in additional security every time. It is a race”*



*Richard ‘Dickie’ George
Technical Director, NSA*

Balancing what (contd.)

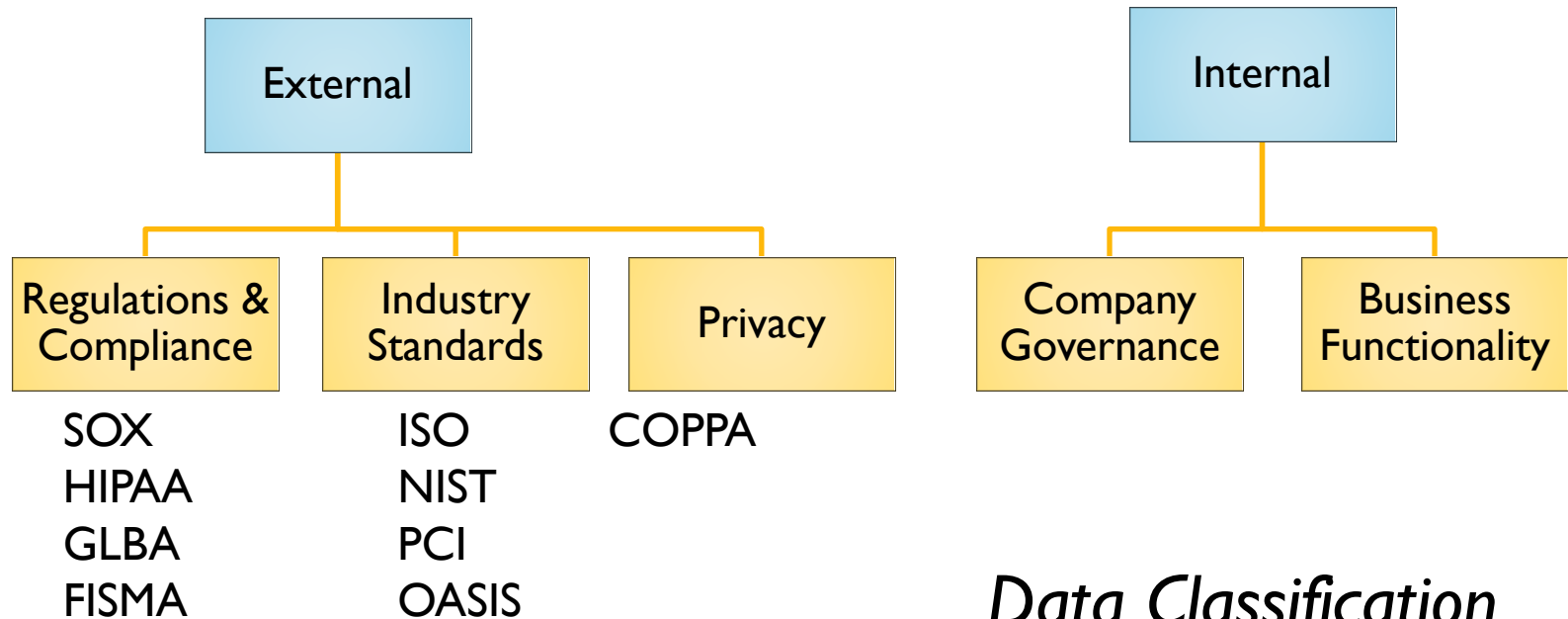
- Threats and Controls

| S.No. | Threat | Control(s) |
|-------|-----------------|---|
| 1 | Overflow | strlen <= bytesize, safe APIs ... |
| 2 | Injection Flaws | Parameterized Queries, Validate input ... |
| 3 | XSS | Response Encoding, Validate Request ... |
| 4 | CSRF | Session specific tokens, POST vs. GET ... |
| 5 | DoS | Load Balancing, Replication ... |
| 6 | Repudiation | Logging, Code signing ... |
| 7 | Reversing | Obfuscation, IsDebuggerPresent API |

#5 – Incorporates Security Requirements

- Lost in translation
 - Send reinforcements, we're going to advance.
 - Send three and four pence, we're going to a dance.
- Seek First to understand, then to be understood

Security Requirements



Data Classification
Subject-Object Matrix
Use / Abuse Case Modeling

#6 – Is Collaboratively Developed

- There is no 'I' in Team
- Synergize

Whose viewpoint?



#7 – Is Adaptable

- The shark is a Polyphyodont
- Sharpen the Saw



Adaptable Software

- Law of resiliency degradation
- Adaptable to
 - Technology
 - Threats
 - Talents
- Begin with the Future in mind
 - Predictive not just proactive

More information Questions?



**Book
Signing**

Cont@ct!

```
If You (Liked the presentation ||
      Did not like the presentation ||
      Need Encore(other) presentation for your company ||
      Have Security Program Development Consulting Needs ||
      Have Security Product Development/Evaluations Needs ||
      Have Awareness, Training & Education Needs ||
      Have Certification Needs)

{
    Contact me;
}
else
{
    Have a great day!
}
finally
{
    Thankyou();
    BuildHighlySecureSoftware();
}
```

- LinkedIn
- Facebook
- Twitter (@manopaul)
- Email
 - mano(dot)paul(at)securisksolutions(dot)com
 - mano(dot)paul(at)expresscertifications(dot)com

