Own your Software Supply Chain

Or it will own you

John Martin, CISSP, CISM

Program Manager, The Boeing Company

John founded Boeing's Commercial Software Security program, cross- and hybrid-cloud security monitoring programs, and Software Security Champions program along with co-founding the Red and application assessment teams, initiated the DevSecOps efforts, and continues being a disruptive influence.

His career spans the years between Blue-Box MF generators, through the era of automated hacks, and into our modern age of industrialized paranoia.

He was named SANS Difference Maker in CyberSecurity in 2016.

He is, unfortunately, a frequent speaker on the topic of commercial software security.

The Board: “What’s up with CyberSecurity?”

<u>And your answer should be…</u>

- “Defense in depth is working well. We’re deflecting over a 1,000,000 automated attacks per day
- “Internal software teams are building resilient code with DevSecOps and the Security Champions initiative is really accelerating that
- “However, we think that because APT attackers are well corralled internally, they’ve increased attacks on our software supply chain. Opportunistic and ransomware adversaries are also attacking the supply chain. We’re actively working programs to encourage our suppliers to certify their software development and the code they deliver to us”

61% of companies have experienced a data breach caused by one of their vendors or third parties

Smack the Headlines (January)

South Korea says mystery hackers cracked advanced weapons servers

Juniper Networks has big bug day

Researchers crack open notorious Fancy Bear rootkit

GandCrab

WannaCry Still Lurks on Infected Computers

Moore's Revenge

Twitter 'very sorry' for security flaw

IoT device malware grew 72% in Q3 alone

Critical credential vulnerability

Hackers take 383 million booking records and 5.3m unencrypted passport numbers …

a four-month-long data breach

770,000,000 email addresses and passwords found in data breach

Computer virus cripples Times

Cyber Hacks To Cost Auto Industry $24 Billion

Oracle mega-update day

iPhone apps linked to Golduck malware

new malware-friendly hosting site

Fortune 100 still use flawed software that led to the Equifax breach

The Attack Process

1. Attackers acquire target identification (i.e. your company, your people, your systems)
2. Using fake identities and Bitcoin, the attackers anonymously rent servers around the world. Those remote servers allowed attackers to launch from a network of machines that (they believe) cannot be traced back
3. Attackers design custom methods to spy on the targets…
   They scout online accounts, mobile devices and social media profiles. They're looking for anything that can be exploited to gain access. Any found email addresses are compared to know UID/Password combinations on the dark web
4. **If those methods are not effective or feasible, the attackers move to your 'trusted' software supply chain. The easiest targets are software developers with online accounts…preferably OBA (old but active) to avoid discovery. (Password Spraying using known UID/pwd combinations)**
5. Attackers employ a team of software developers to identify and build (or reuse) appropriate computer attacks for those specific devices or accounts used by the targets
6. Attackers employ an operational team to breach each specific target. The initial goal is to breach and establish deep presence, often employing custom malicious software on the targets' systems, other times using the targets own security systems, to maintain access
7. Attackers begin to exfiltrate information (slowly). Information retrieved is decrypted, organized and analyzed
8. After the attackers achieve access to your stuff, they'll maintain surveillance your people and continue to vacuum up emails, photos and locations for as long as possible (or until it become financially unfeasible).

Attackers always have three broad avenues to attack: software, configurations, people.

Note: When a software supplier is compromised, the code delivered to you MUST be assumed to be compromised. Additionally, any instructions or configuration information MUST be assumed to be compromised.

Causes and Conditions

Code

- 80% of organization fail to test applications for security in the SDLC
- 60% of all applications in 2017 were <u>always</u> vulnerable
- Serious vulnerabilities in applications are increasing at a rate of 10%
- Microservices have <u>many</u> more vulnerabilities per line of code than traditional applications
- >90% of applications include third party and open source components creating a new level of inherited risk
- >25% of software vulnerabilities are inherited from third party and open source
- 80% of software developers aren't familiar with SAFECode.org Fundamental Practices

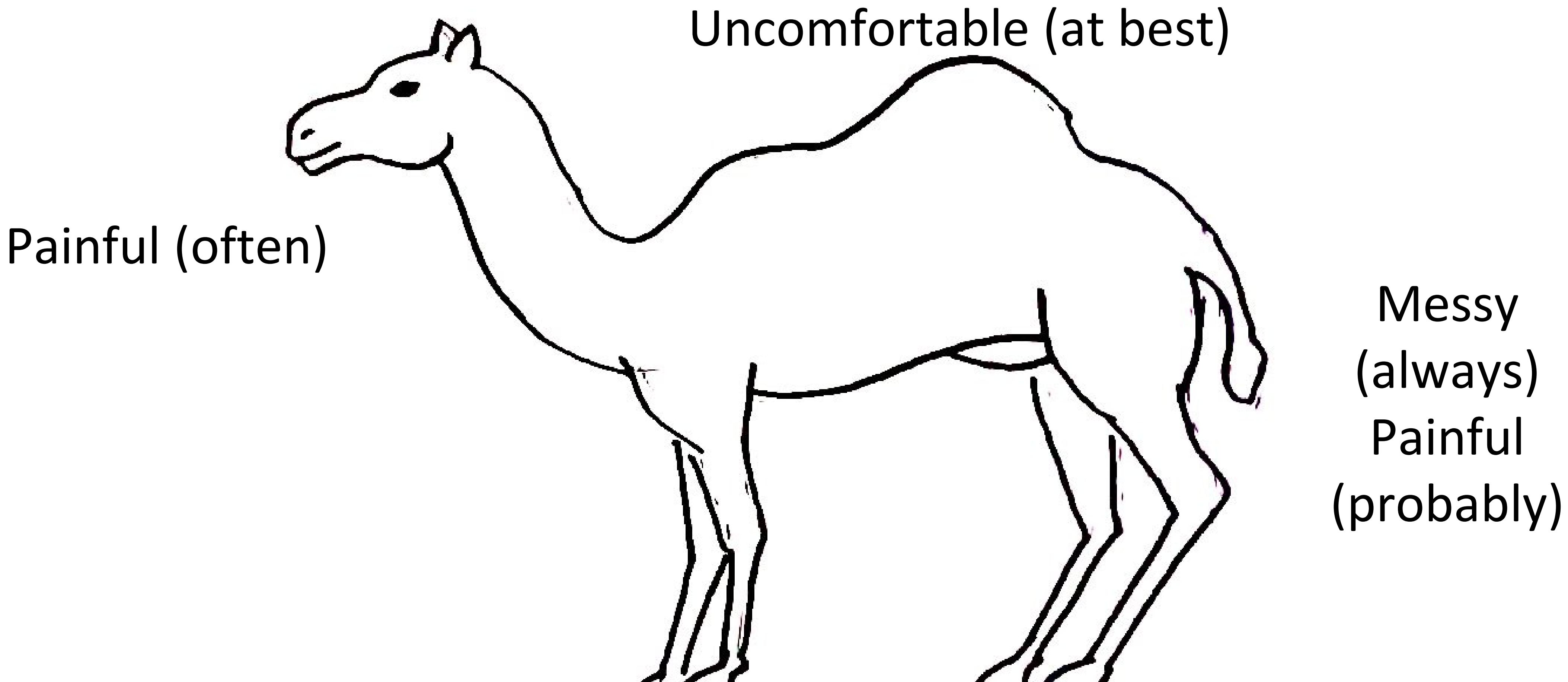
Configurations

- 1.5 billion sensitive files exposed by misconfigured servers/services; 16% of total
- 80% of companies still use passwords to protect servers; 15% of servers are unsupported
- 22% of servers/services have at least one critical vuln

People

- 1.5 billion sensitive files exposed by misconfigured servers/services; 16% of total
- 80% of companies still use passwords to protect servers
- 45% of system designers aren't familiar with CIS Controls

Camel Model of Software Supply Chain Security



Facts

Industry-wide, Software is Getting Better (but it’s still horrible)	
Percentage of Applications with Critical Defects at 1st Test among small & mid-tier suppliers	
2010 - 2015	96% Fail Rate
2016 - 2017	80% Fail Rate
2018 -	71% Fail Rate
	*major contributors on test fail rates include Boeing, Veracode, Foundstone, Synopsis

What Small Suppliers Say

- “We use industry best practices!”
- “Customers have a labyrinth of requirements that don’t make sense”
- “No clear way to engage customer’s security team”
- “Security is like Dr. No. They come in, tell us NO, then disappear”

“Customers are like Helicopter parents, they hover over us, tell us to do it better, then disappear”

Basic buyer assumption

If you’re selling us stuff,
you’re responsible for the stuff you’re selling

Basic seller problem

Get the income today,
solve the problems tomorrow

But it’s not that easy

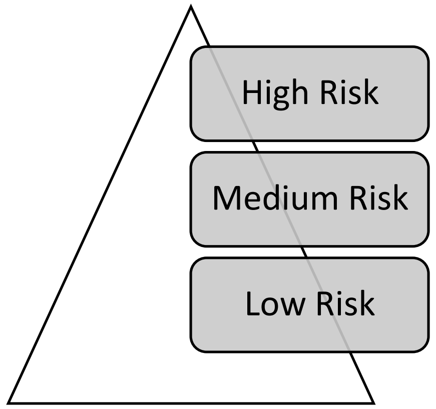
Buyer’s Issues

- Bad Code
  (starts with components)
- Vulnerable implementations
  (which one, ones?)
- Too much stuff
  (what’s important?)
- Poor auditability
  (inventories & other gaps)

Mature Seller’s Issues

- Can’t provide all customers with the (sometimes silly) information they require to make purchase decisions
- No agreement on what information might mitigate actual/perceived risks
- Many requests do not align well with real-world secure development practices

So, What to Do?



- Establish a common language for software security
- Inventory and risk-rank existing software
  - Top 10% (later top 20%)
- Begin contacting existing vendors…let them know requirements are coming; let them know you’re watching
- Perform Threat Models against the reference architecture for New Purchases
- Insert correct language into the RFI and RFP documents
- Final contract acceptance clause
- About Commodity Software
  - General function software
  - Top 100 companies

Build the Communications and Contracts

Request for Proposal (or Information)

- **Rule:** Software must be validated to be free from significant security defect prior to acceptance.
  Describe the process (e.g. ISO 27034) and methods (e.g.3rd-party code review) used to ensure that the product is delivered free from significant security vulnerabilities.
- **Rule:** Vulnerability management of zero-day attacks to installed code, including 3rd-party code, is a priority.
  Describe the processes and methods used to ensure that the product (including any 3rd-party code) remains free from significant vulnerabilities throughout its lifespan.

Software Acceptance Clause (no surprises)

a. that the Goods contain no defects that exceed a Common Vulnerability Scoring System ("CVSS") score of 6.0, as assessed by a third party assessment organization approved in writing by Buyer;

OR

b. that Seller's secure development lifecycle is in substantial alignment with ISO 27034.

When all this doesn’t work…

Establish a fact-based trust

- this often means helping the supplier to evolve toward better practices

Or not…

Or buy yet more expensive 3rd-party wrappers to enclose the high risk software with less-risky wrappers…but be sure to ask the ‘wrapper’ companies these same questions about THEIR software

Recap

Know what you need

- Inventory
- Discover risk
- Act on the risky bits
- Avoid bland assurances

Get (most of) what you want

- Common language
- Clear RFI/RFP rules
- 2-party Acceptance Clause

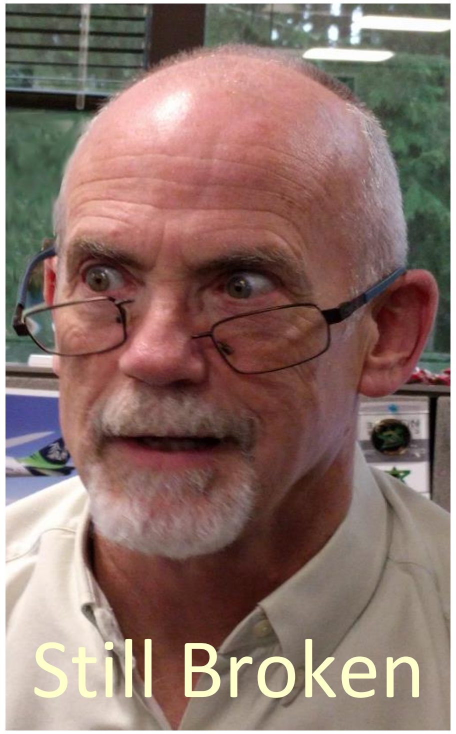
You’ll probably have to help vendors evolve upward

A few questions

- Will digital signatures/block-chain help?
- Can this be part of my Security-as-a-Service solution?
- How does this solve my configurations problem?

End

But NOT The End



ISO/IEC 27034 (7 part series)

From a buyer's perspective

- Offers a concise internationally recognized way to get transparency into a supplier's software security management process.

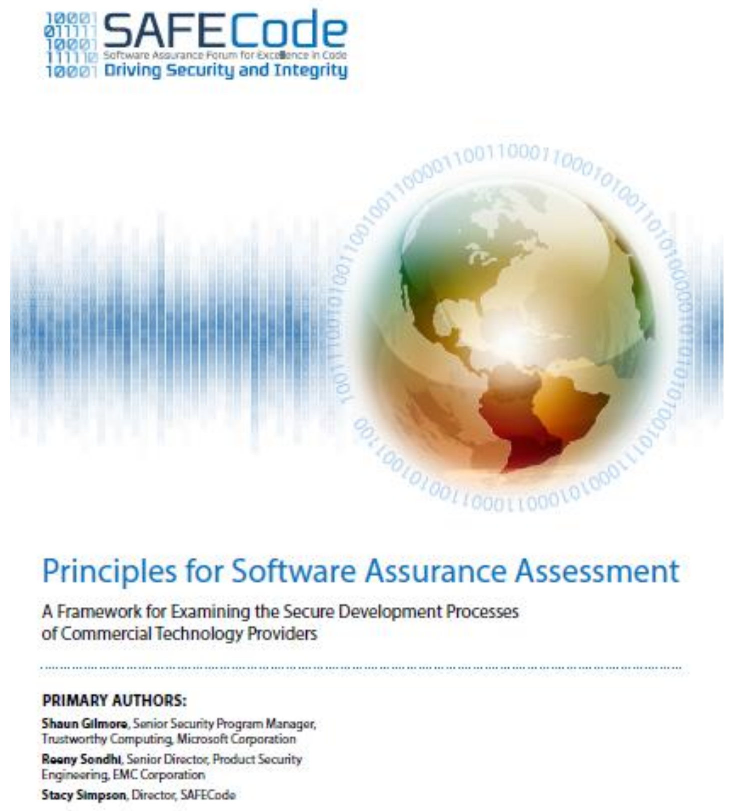
From a supplier perspective

- Offers a single answer to demand for software security process assurance
- 27034 is flexible enough to align with diverse engineering organizations
- Specific and rigorous enough to address real world risk

Principles of Software Assurance Assessment

- No “one-size fits all”
- Tiered approach based on supplier maturity

1. Less mature suppliers: Assessment will be primarily a tool-based, testing-centric approach
2. More mature suppliers: Focus should be on a process-based assessment
3. Where applicable and available, international standards should be the default assessment approach



SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

Principles for Software Assurance Assessment

A Framework for Examining the Secure Development Processes of Commercial Technology Providers

PRIMARY AUTHORS:

Shaun Gilmore, Senior Security Program Manager, Trustworthy Computing, Microsoft Corporation

Reeny Sondhi, Senior Director, Product Security Engineering, EMC Corporation

Stacy Simpson, Director, SAFECode

http://www.safecode.org/publications/

Still the End

But NOT The End