



Threat Modeling

"Threat modeling at the design phase is really the only way to bake security into the SDLC." – Michael Howard, Microsoft

Nathan Sportsman

Founder and Chief Executive Officer

Agenda

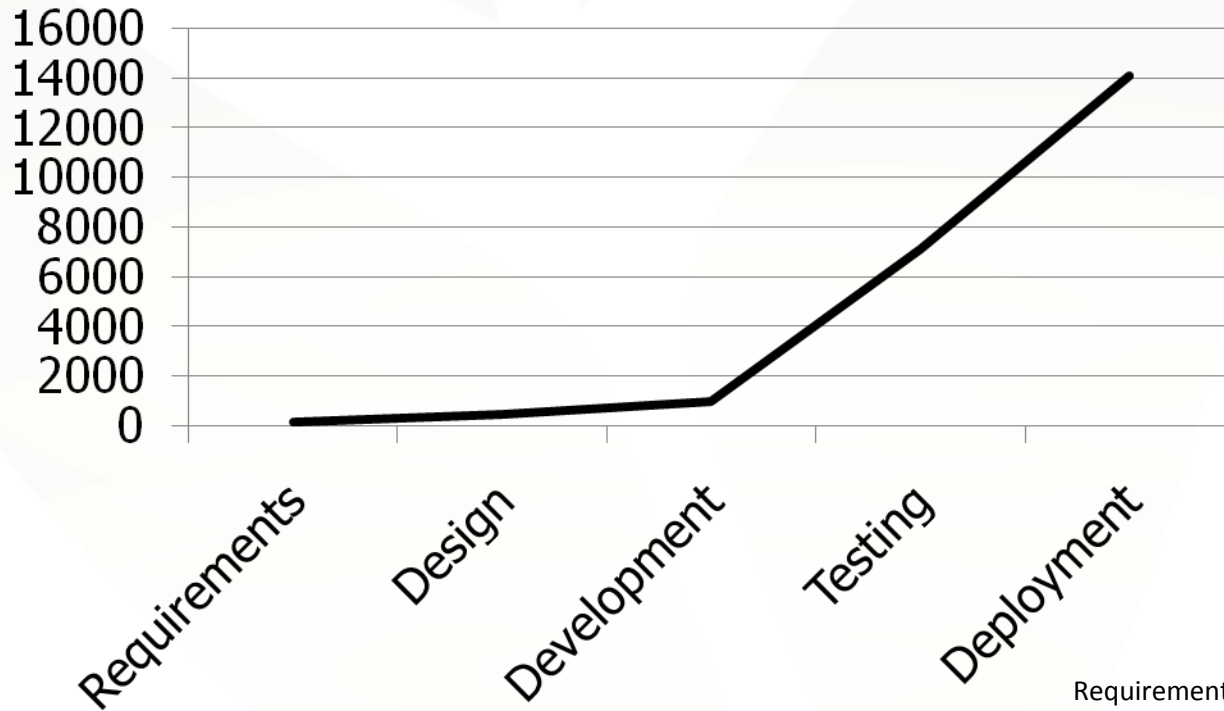
- ❖ Introduction
- ❖ Process Overview
- ❖ Current State Analysis
- ❖ Workshop

INTRODUCTION

Secure Development Lifecycle

Activities	Core	Security
Planning		
Requirements and Analysis	Functional Requirements Non Functional Requirements Technical Requirements	Security Objectives
Architecture and Design	Design Guidelines Architecture and Design Review	Threat Modeling Architectural Risk Analysis
Development	Unit Tests Code Review Daily Builds	Security Code Review
Testing	Integrated Testing System Testing	Security Testing
Deployment	Deployment Review	Penetration Testing
Maintenance		Risk Assessment

Cost Per Defect



Estimates provided by IEEE Computer Society

Requirements	\$139
Design	\$455
Development	\$977
Testing	\$7,136
Deployment	\$14,102

Threat Modeling Is

- ❖ ...a security control performed during the architecture and design phase of the SDLC to identify and reduce risk within software
 - » Also called architectural risk analysis depending what camp you talk to
 - » Sit down between security experts and architects & development leads
 - » Performed through white boarding and thoughtful discussion
 - » Accomplished over 2 to 5 day time period depending on application complexity

Threat Modeling Benefits

❖ Improves Security

- » Champions threat analysis
- » Uncovers logical/architectural vulnerabilities
- » Reduces risk and minimizes impact

❖ Drives Testing

- » Validates design meets security requirements
- » Reduces scope of code inspection
- » Serves as a guide for verification testing

❖ Reduces Cost

- » Identifies expensive mistakes early on
- » Improve understanding and structure of application
- » Decreases new hire ramp up time

THREAT MODELING PROCESS

Threat Modeling Approaches

❖ Attack Centric

- » Evaluates from the point of view of an attacker

❖ Defense Centric

- » Evaluates weakness in security controls

❖ Asset Centric

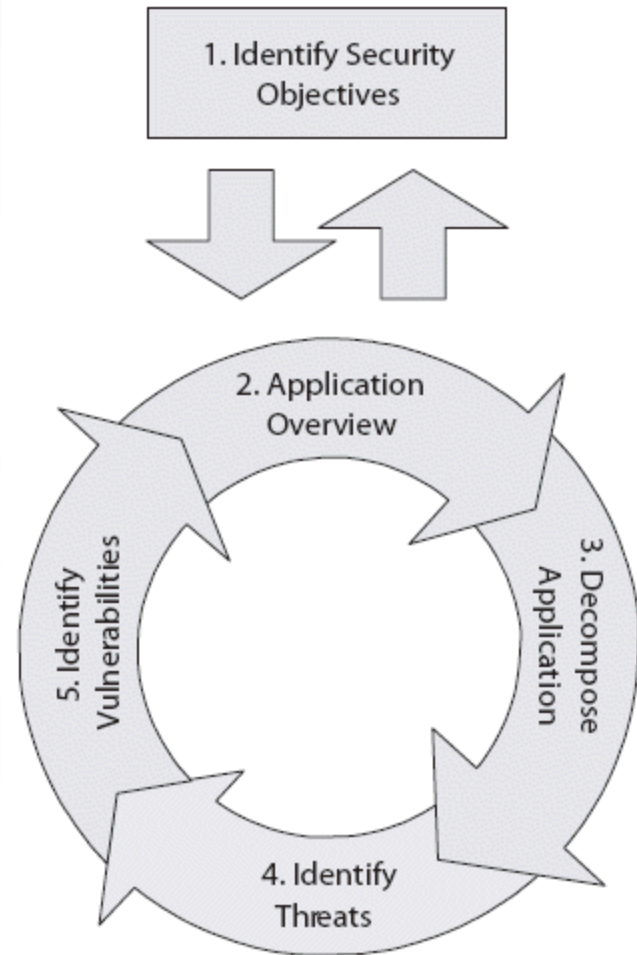
- » Evaluates from asset classification and value

❖ Hybrid

- » Evaluates application design using combination of methodologies to meet security objectives

Threat Modeling Process

- » Identify Security Objectives
- » Application Overview
- » Decompose Application
- » Identify Threats (and Countermeasures)
- » Identify Vulnerabilities
- » Repeat



APPLICATION UNDERSTANDING

Application Overview

- ❖ Review any related documentation provided before the threat model
 - Documentation is usually out of date, but can still provide an overview of the application
- ❖ Developers should give an overview of the applications purpose and key features
- ❖ Review security requirements and how the design fulfills those specifications

Characterize The System

- ❖ Scenarios are very useful in identifying threats
- ❖ Common use and abuse cases should be understood
- ❖ Cases should come from security requirements, If cases are not available, create and analyze your own abuse cases with developers

APPLICATION DECOMPOSITION

Application Modeling

❖ Flow Chart

- » Less commonly used
- » Difficult to overlay threats

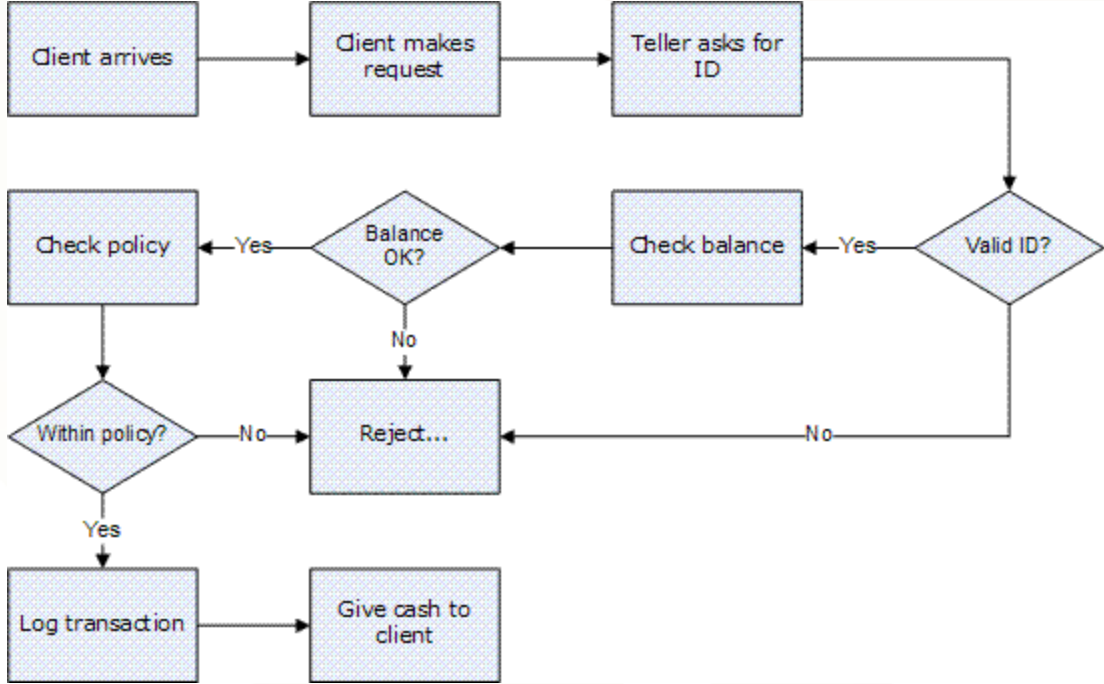
❖ Data Flow Diagrams (DFD)

- » Hierarchical in nature
- » Focused on data input
- » Representation used by Microsoft

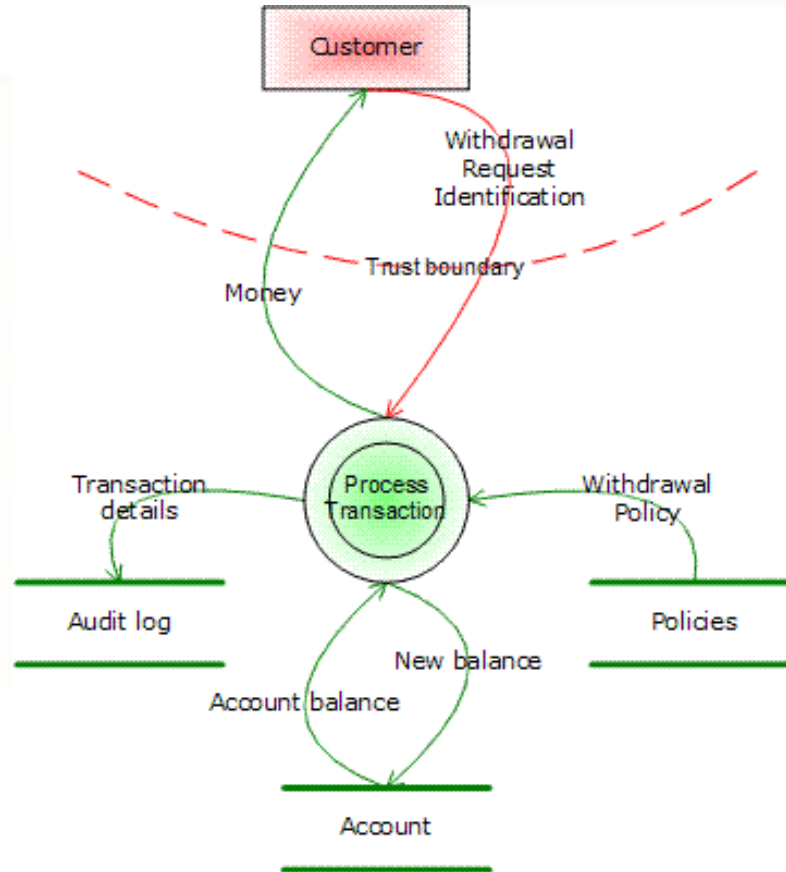
❖ Universal Modeling Language (UML)

- » Familiar to developers
- » OO and component diagramming

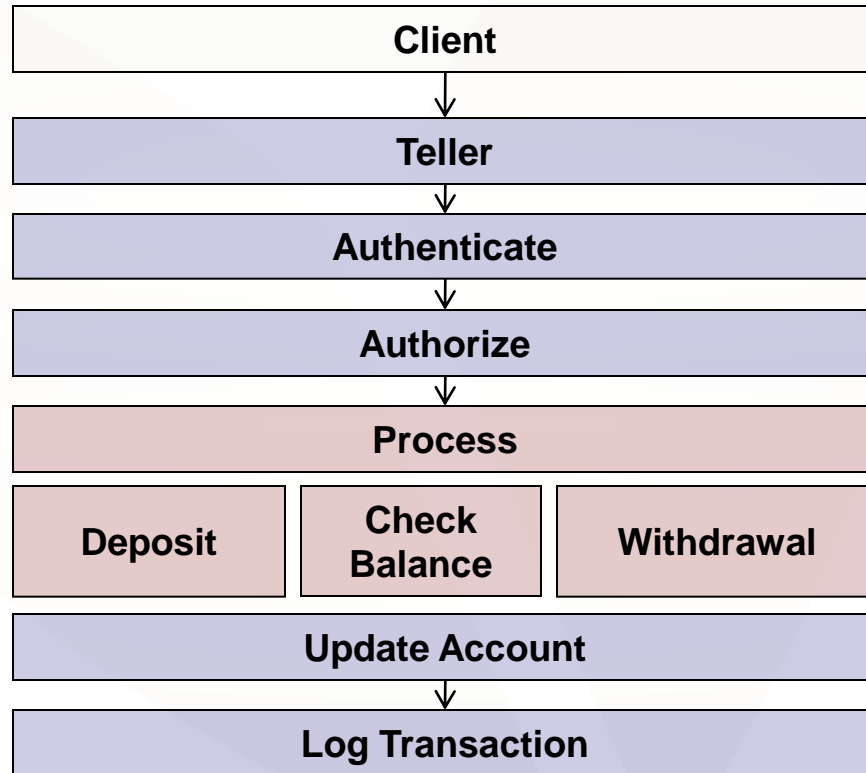
Flow Chart



Data Flow Diagram



Universal Modeling Language (Functional)



THREAT IDENTIFICATION

Microsoft STRIDE Categorization

Spoofing

- Example: Session identifiers are incremental. Another users session id can be guessed and used

Tampering

- Example: Data validation of user input does not occur. Database entries can be modified via SQL injection

Repudiation

- Example: System does not have audit functionality of user operations to trace an improper request

Information Disclosure

- Example: Verbose error messages reveal database schema

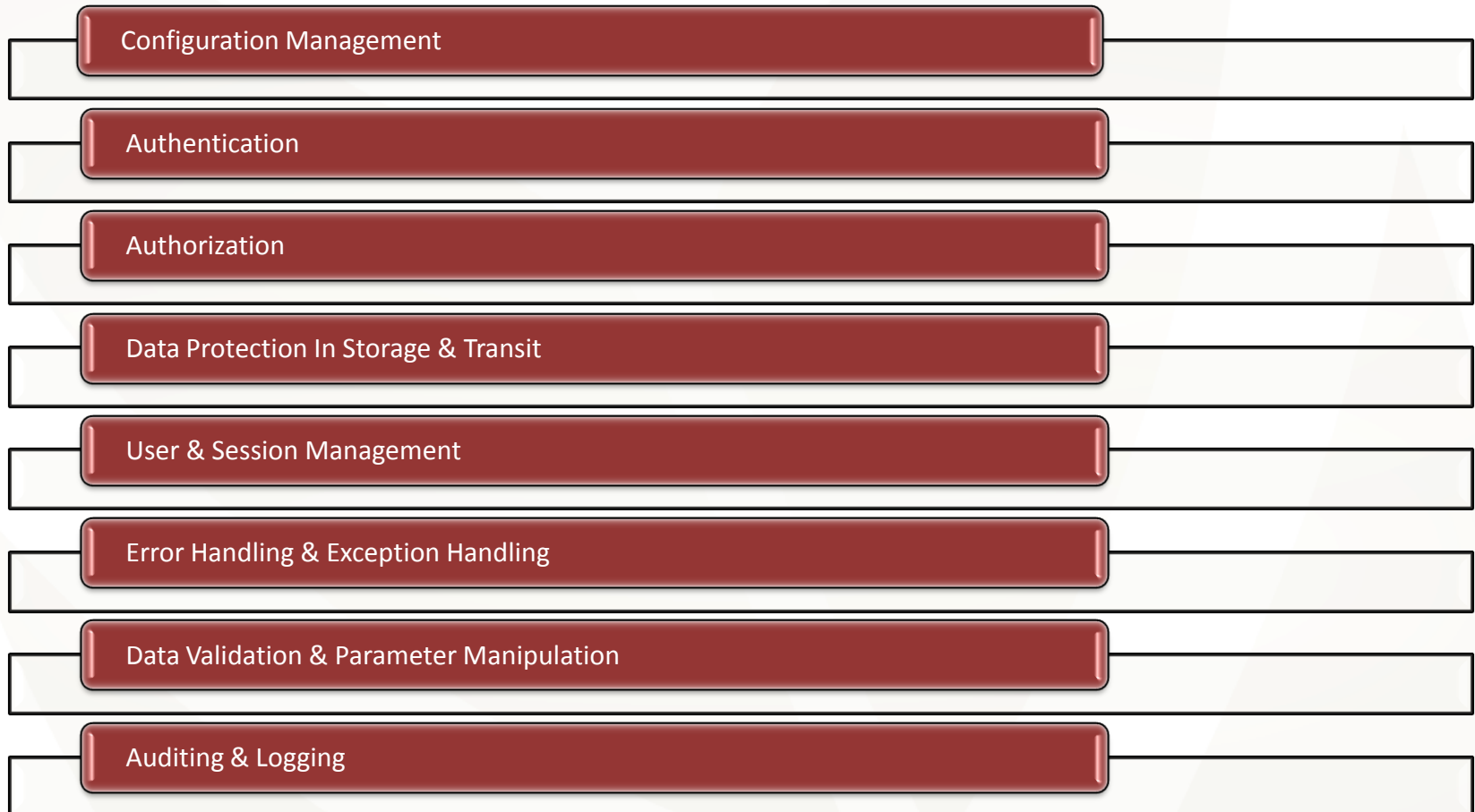
Denial of Service

- Example: Application crashes from unexpected user input

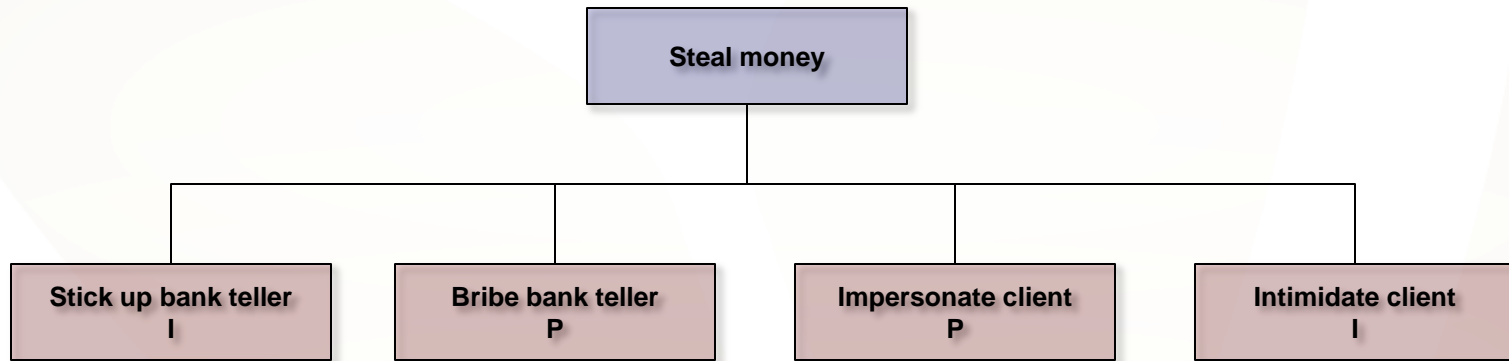
Elevation of Privilege

- Example: User is able to obtain administrative access to application through variable change

Application Security Frame Categorization



Threat Trees



P – Probable
I - Improbable

VULNERABILITY PRIORITIZATION

General Risk Approach

$$Risk = Threat \times Vulnerability \times Cost$$

Threat = An action intended to do damage or harm

Vulnerability = Likelihood of success of a threat

Cost = Total impact or cost of the incident

Microsoft DREAD Categorization

Damage

- How much damage would occur if the attack was successful

Reproducibility

- How difficult is it to reproduce the attack

Exploitability

- What is the likelihood the threat can moved from a theoretical attack to a functional exploit

Affected User

- What percentage of the user would be affected if the vulnerability was exploited

Discoverability

- How difficult would it be for an attacker to discover the vulnerability

Structured Lists

- ❖ Round table discussion
- ❖ Prioritization based on consensus
- ❖ Utilize Praetorian Risk Approach
- ❖ Numbered list generated

CURRENT STATE ANALYSIS

Activities Organizations Are Doing Now

- ❖ Only a handful of companies are actually introducing security into the SDLC in any meaningful way
- ❖ Prior focus by vendors, security service providers, and clients tended to be on back end security controls
- ❖ Other than industry thought leaders, little effort is being paid to the importance of front end controls such as security requirement reviews and threat modeling
- ❖ Backwards approach due to lack of awareness, preexisting systems, and legacy code

Adoption Constraints

- ❖ In its current form threat modeling generally requires outside security expertise
- ❖ Expensive cost limits threat modeling to one off expenditures for only the most critical applications
- ❖ Difficult to internalize and reproduce process across application portfolios
- ❖ Tools limited functionality and to scalability inhibits development team empowerment

Future Predictions

- ❖ Software security initiatives at more and more organizations will mature and incorporate threat modeling
- ❖ Threat modeling tools will become more sophisticated and enterprise ready

References

- ❖ Frank Swiderski and Window Snyder. Threat Modeling. Microsoft Press, 2004.
- ❖ Michael Howard and David LeBlanc. Writing Secure Code 2nd Edition. Microsoft Press, .
- ❖ Microsoft Application Threat Modeling Blog
<http://blogs.msdn.com/threatmodeling/>
- ❖ Microsoft SDL Threat Modeling Tool
<http://msdn.microsoft.com/en-us/security/dd206731.aspx>
- ❖ Building Maturity in Security Model
<http://www.bsi-mm.com/>
- ❖ OWASP Application Threat Modeling
http://www.owasp.org/index.php/Application_Threat_Modeling

Unlock Conundrum Workshop

- ❖ Cell Phone Manufacturer is caught with its pants down and has found out that \$6,000,000 worth of unlock codes have been sold by their overseas support personnel
- ❖ Using a internal, web based tool, support personnel are able to query unlock codes used to unlock the phone for diagnostic purposes
- ❖ Unfortunately, employees are abusing the system and setting up websites and selling unlock codes out of band to customers for personal profit
- ❖ Manufacturer has asked a threat model be performed on the new design of the application which they hope will stop the abuse

Threat Modeling

“In fact, during the Windows Security Push (and all pushed that followed at Microsoft, we found that the most important aspect of the software design process, from a security viewpoint, is threat modeling.”

– Michael Howard, Microsoft, Writing Secure Code 2nd Edition

Nathan Sportsman

Founder and Chief Executive Officer