Threat Modeling
A Brief History and Unified Approach at Intuit

OWASP
The Open Web Application Security Project

About Us

OWASP
The Open Web Application Security Project

- Tin Zaw
  - Staff Software Engineer
  - Intuit, Inc

- Scott Matsumoto
  - Principal Consultant
  - Cigital, Inc

Agenda

OWASP
The Open Web Application Security Project

- Background
  - What Threat Modeling means (to us)
  - What our TM program looks like
- System Threat Modeling
- Lessons and Takeaways
- Protocol Threat Modeling
- Concluding Remarks
- Questions

OWASP
The Open Web Application Security Project

BACKGROUND

Threat Modeling

OWASP
The Open Web Application Security Project

- Threat Modeling is a software design analysis that looks for security weaknesses by juxtaposing software design views against a set of attackers.
  - It identifies secure-design weaknesses
    - Missing security controls
    - Weak or inappropriate security controls
    - Potential vulnerabilities
  - Finds weaknesses that cannot be found by other techniques
  - It is not a replacement for Pen-Testing or Secure Code Review

Unified Threat Modeling Program Goals at Intuit

OWASP
The Open Web Application Security Project

- Create a single approach from three disparate approaches in practice:
  - STRIDE
  - Homegrown1
  - Homegrown2
- Develop a common process so one security engineers can share work
- Develop a common process that would allow the business units do the threat models
- Create a shared vision of the security concerns with development early in the SDLC

Unified Threat Modeling Program Elements

OWASP
The Open Web Application Security Project

1. Definition of the process
2. Formal training for security and the business units
3. Introduction of metrics
4. Scaling the new approach through tools and other automation

Unified Threat Modeling Process

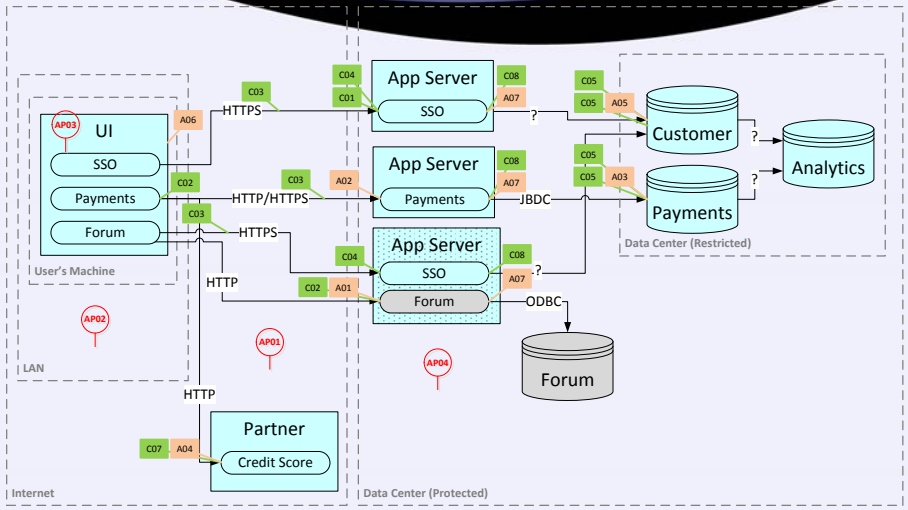
OWASP
The Open Web Application Security Project

- Define scope and depth of analysis
- Gain understanding of what is being threat modeled
  - Use existing development artifacts
- Model the threat structure
  - Identify Assets, Security Controls and Attackers
  - Juxtapose threat structure and software model
- Interpret the threat model
  - Produce the list of threats
- Create the threat table for reporting the threats
  - Rank the risk of the threat
  - Propose mitigations

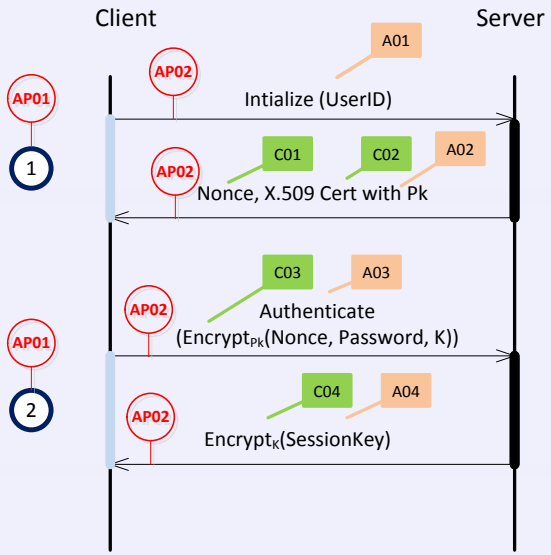
Different Types Of Threat Models

OWASP
The Open Web Application Security Project

- System Threat Model
  - A holistic view of an application's security posture
  - Best view the security between the application and infrastructure
  - Good for building a roadmap for additional security activities



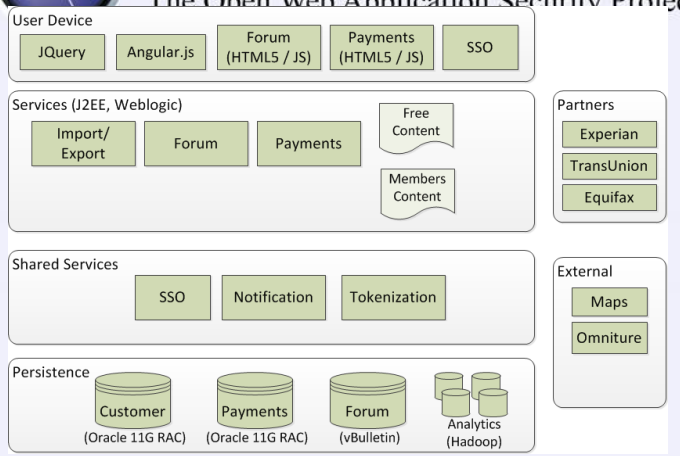
- Protocol Threat Model
  - Analysis of message structure and interaction between components
  - Bridges the gap between a System Threat Model and Code Review



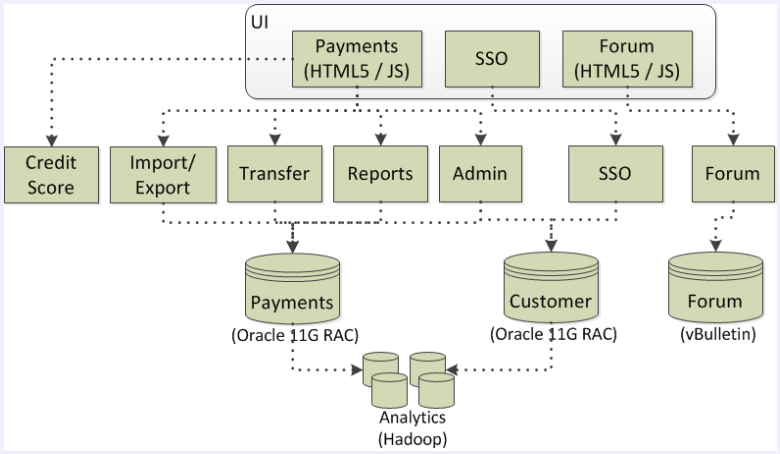
OWASP
The Open Web Application Security Project

SYSTEM THREAT MODELING

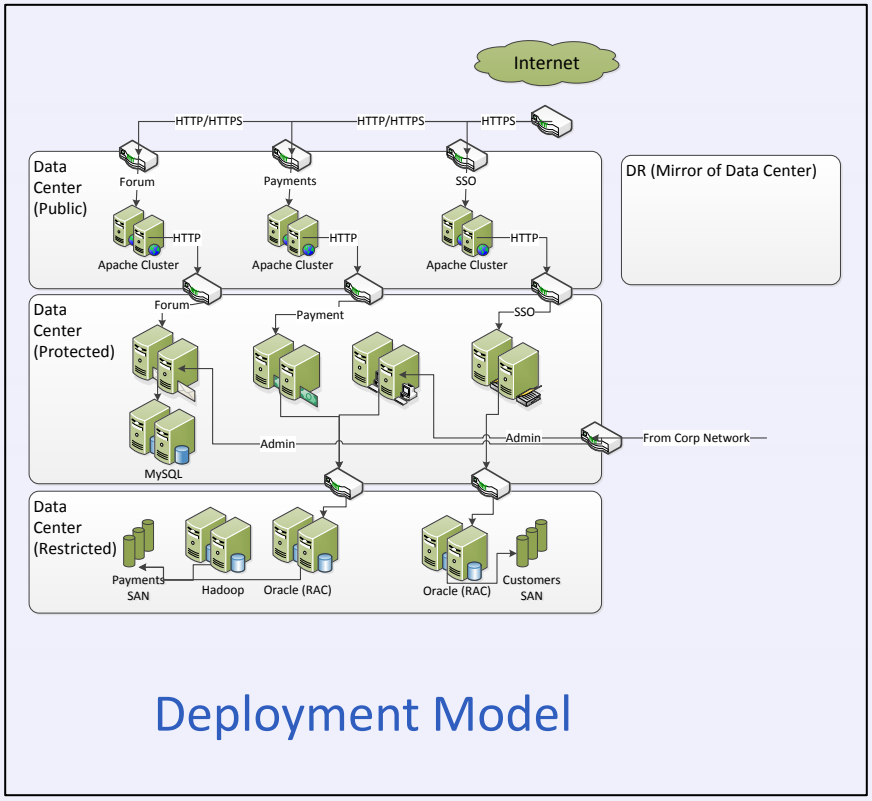
Leverage Development Generated System Analysis



Layer Model

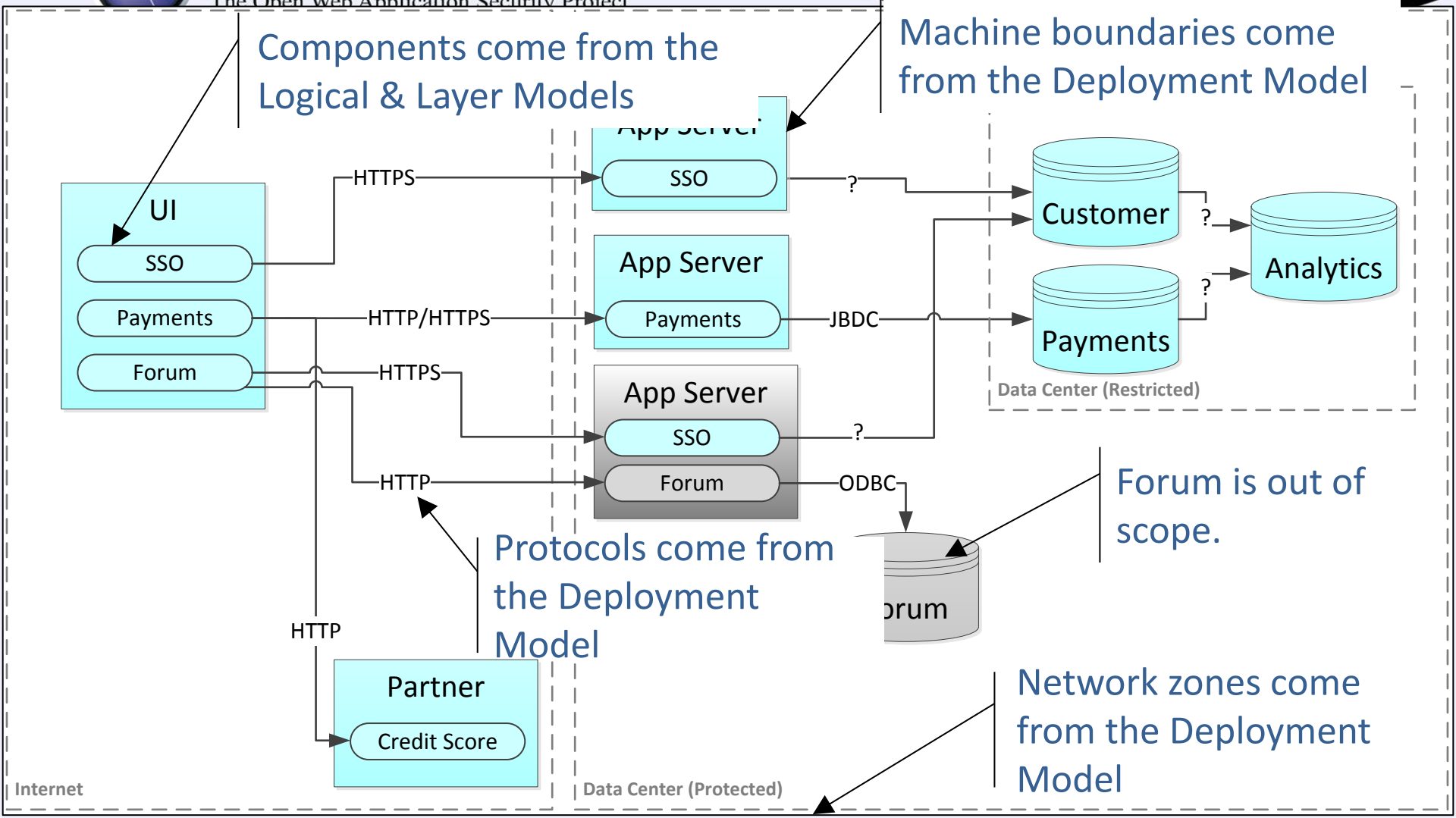


Logical Model



Deployment Model

Create a Simplified System Model



Modeling the Threat Structure

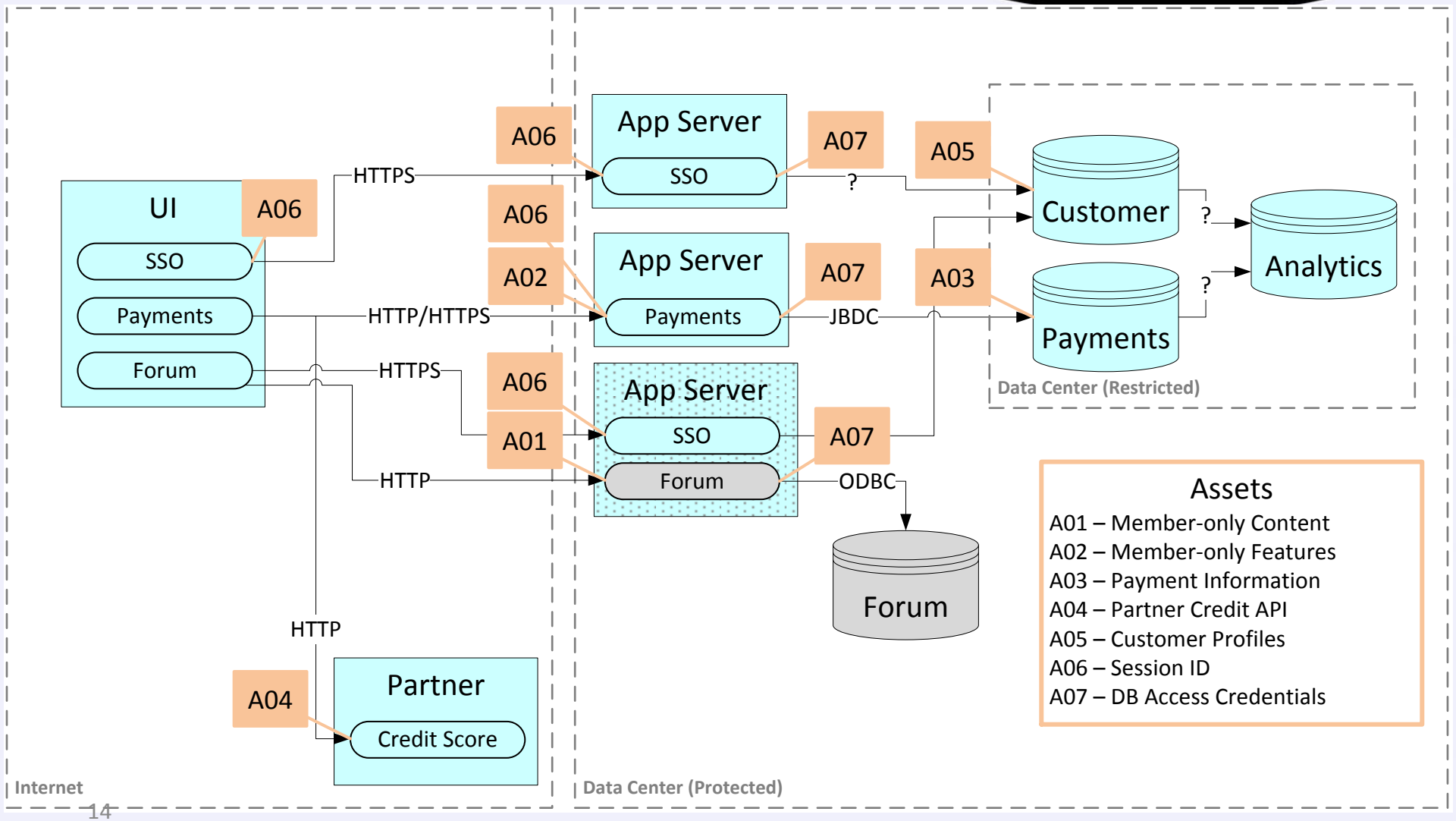
OWASP
The Open Web Application Security Project

- We continue to analyze the information we've collected in our interviews and now add the threat related elements.

Assets	The data and functions that the system must protect
Security Controls	The mechanisms currently designed and implemented to protect the Assets
Attacker Profiles	The actors what want to harm the system

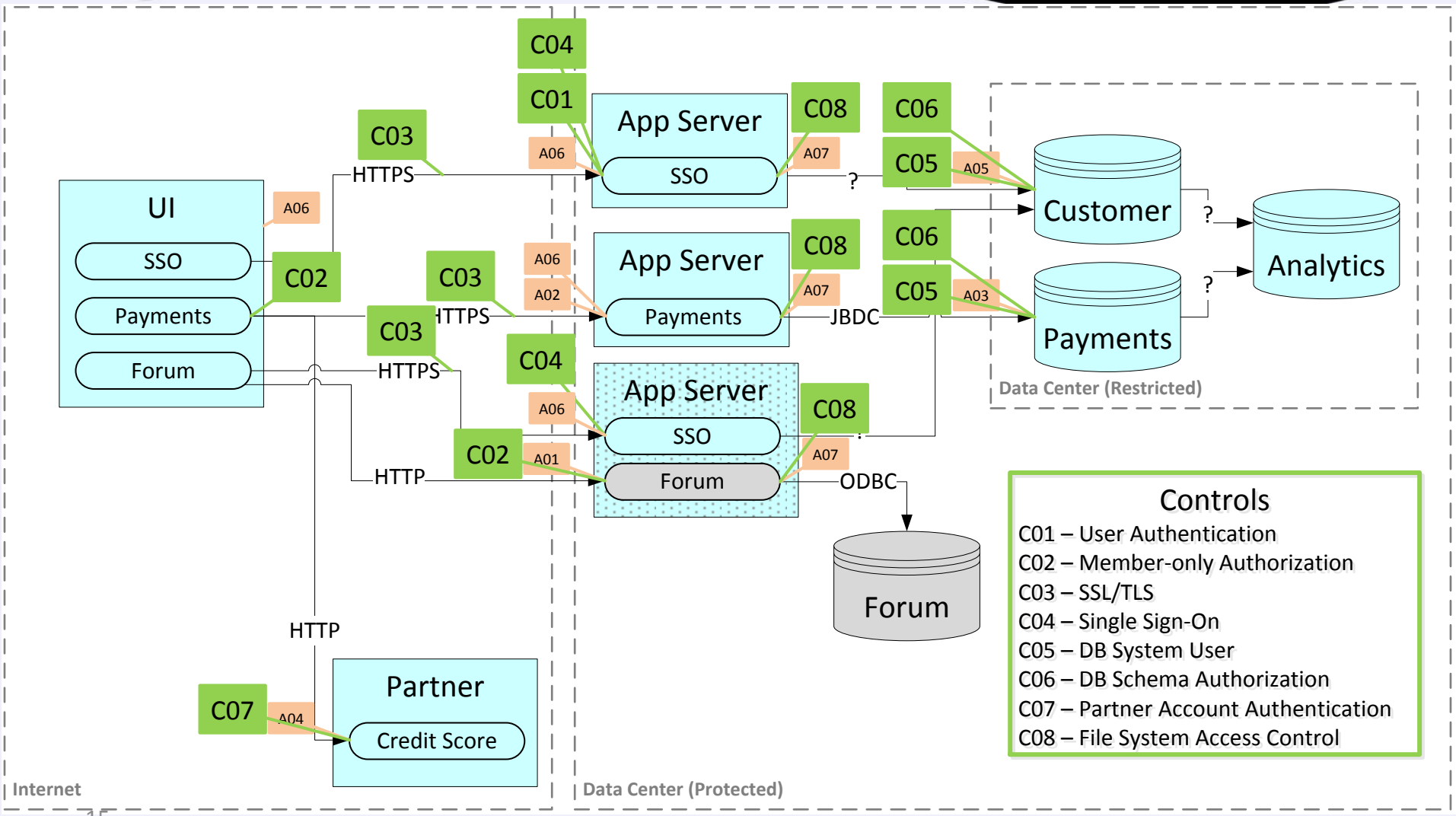
- Juxtaposing the Threat Structure and the System creates the actual Threat Model. Interpreting the model produces a list of potential threats.

Model The Threat Structure - Assets



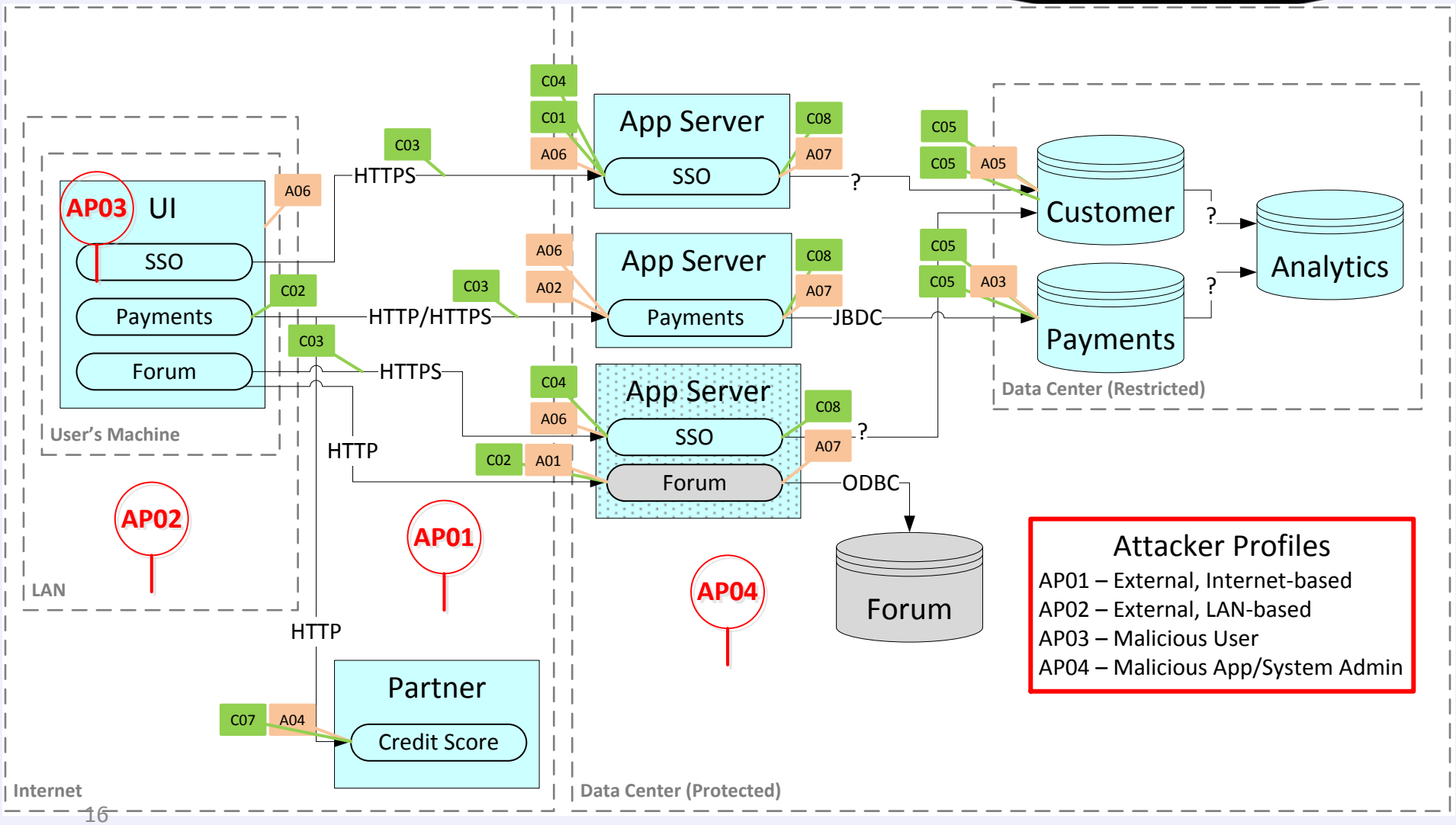
Model The Threat Structure – Security Controls

OWASP
The Open Web Application Security Project



Model The Threat Structure – Attacker Profiles

OWASP
The Open Web Application Security Project



Interpret The Threat Model

OWASP
The Open Web Application Security Project

- Using the model, start with an Attacker and follow the flow-of-control paths in the system to reach an Asset
  - Is there any path where Attacker can reach Asset without going through a Control?
  - For any Security Control along each of those paths:
    - What must the Attacker do to defeat the Control?
    - Can Attacker defeat the Control?
- Record missing or weak controls in the Threat Table

Create the Threat Table

OWASP
The Open Web Application Security Project

- Collect Threats into the Threat Table.

- Each entry in the threat table:
  - Identifies the threat
  - Calculates the risk based on the attacker profile and the existing controls
  - Proposes mitigations to development to reduce the risk to an acceptable level.
    - Mitigations should be practical and implementable
    - Important to create a “shared vision” with the development team

OWASP
The Open Web Application Security Project

LESSONS AND TAKEAWAYS

Lessons and Experiences from the Intuit Journey

OWASP
The Open Web Application Security Project

- Classroom Observations
- Threat Modeling Adoption Challenges
- Threat Modeling Program Wins

Classroom Observations

OWASP
The Open Web Application Security Project

- Self-contained nature of the course
  - Come as you are, no need to do any preparation
- In-class group exercises
  - Problem solving by a group of different backgrounds – architects, Ops, testers, coders
- Mindset difference
  - Developers think of Assets first, Security Engineers think of Attackers first

Threat Modeling Adoption Challenges

OWASP
The Open Web Application Security Project

- The term “Threat Modeling”
  - Foreign language to some, and most have their own interpretation of what it is
- Time commitment
  - Time for entire day of training (esp. when the subject does not seem directly related to their job)
  - Time required to Threat Model
- Getting Development to lead TM exercise
  - So far, TMs get done when Security leads. Need to change

Threat Model Program Wins

OWASP
The Open Web Application Security Project

- Instilling the security mindset
  - Not just for architects and developers. Everyone benefits from the training
- Alignment of security goals
  - “You agreed that it was an asset, so you need to protect it.”
- Flaws discovered exclusively by Threat Modeling
  - Design flaws, logic flaws
- Consistent language, diagrams and document format
  - Among product teams and between product and security teams, company wide

OWASP
The Open Web Application Security Project

PROTOCOL THREAT MODELING

Protocol/Sequence/API Threat Model

OWASP
The Open Web Application Security Project

- Protocol is an agreed-upon sequence of interactions between two or more components
  - Including data/message formats in these interactions

- Why Threat Model a protocol?
  - Analyzing the details of the interactions between components will reveal weaknesses that a System Threat Model will not. For example:
    - Interleaving threats
    - Relay threats

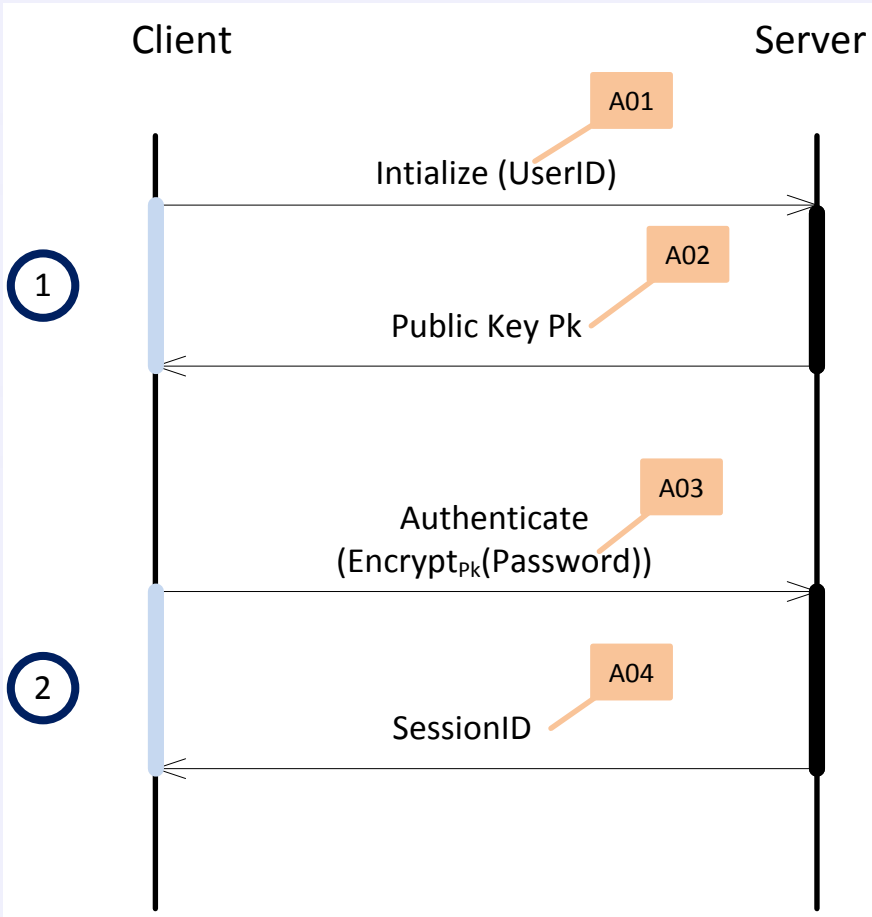
Modeling the Protocol – Attacker Profiles

OWASP
The Open Web Application Security Project

- Two types of canonical attacker profiles
  - Malicious Client – the component action as the client in the interaction*
  - Man-in-the-Middle – any attacker who is able to observe / tamper with the messages in the interaction

- Some protocols may involve more than two entities (e.g. OAuth)
- Role of “client” and “server” may change depending on the interaction.

Model the Protocol - Assets

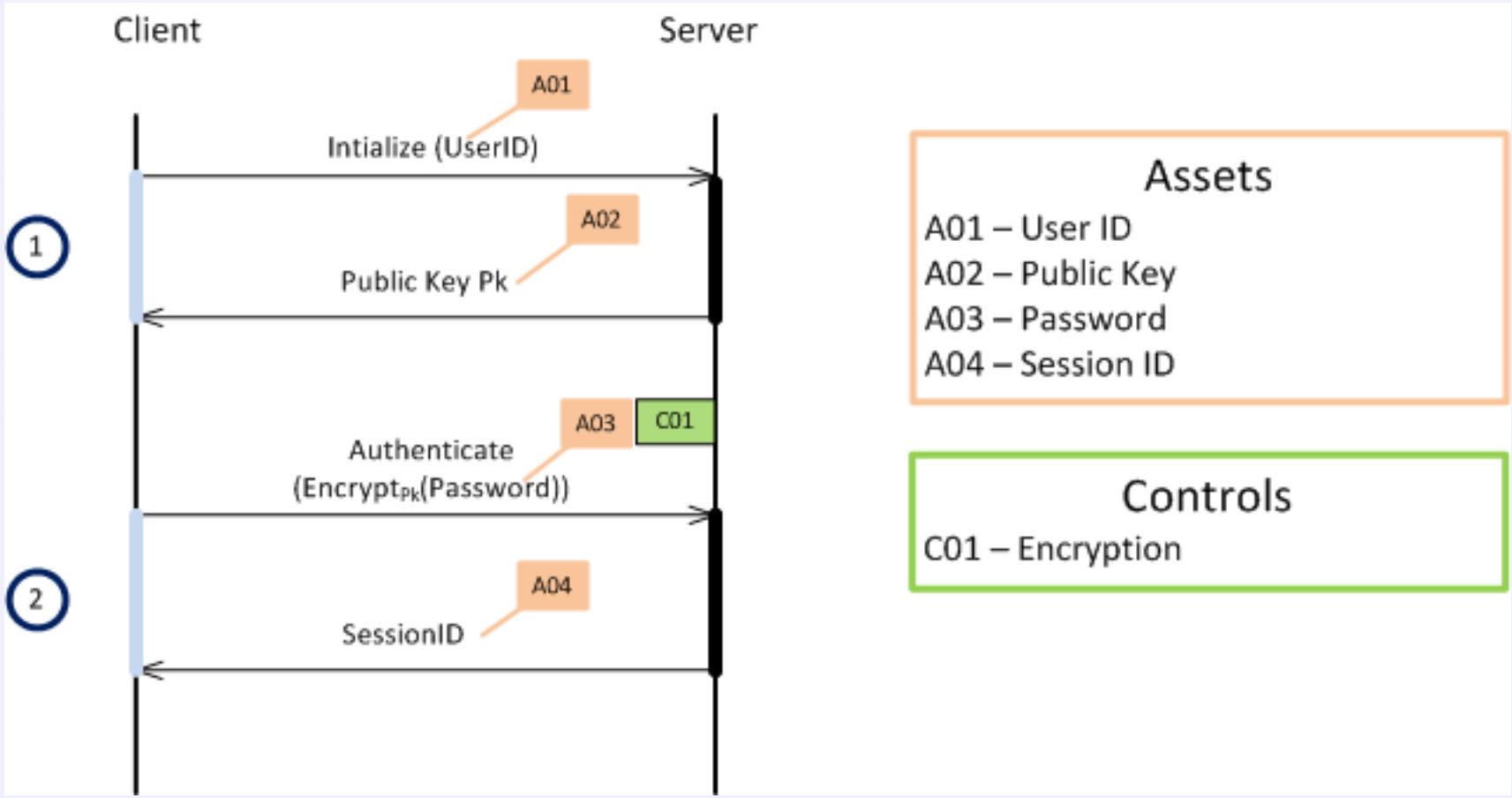


Assets are the data in the messages

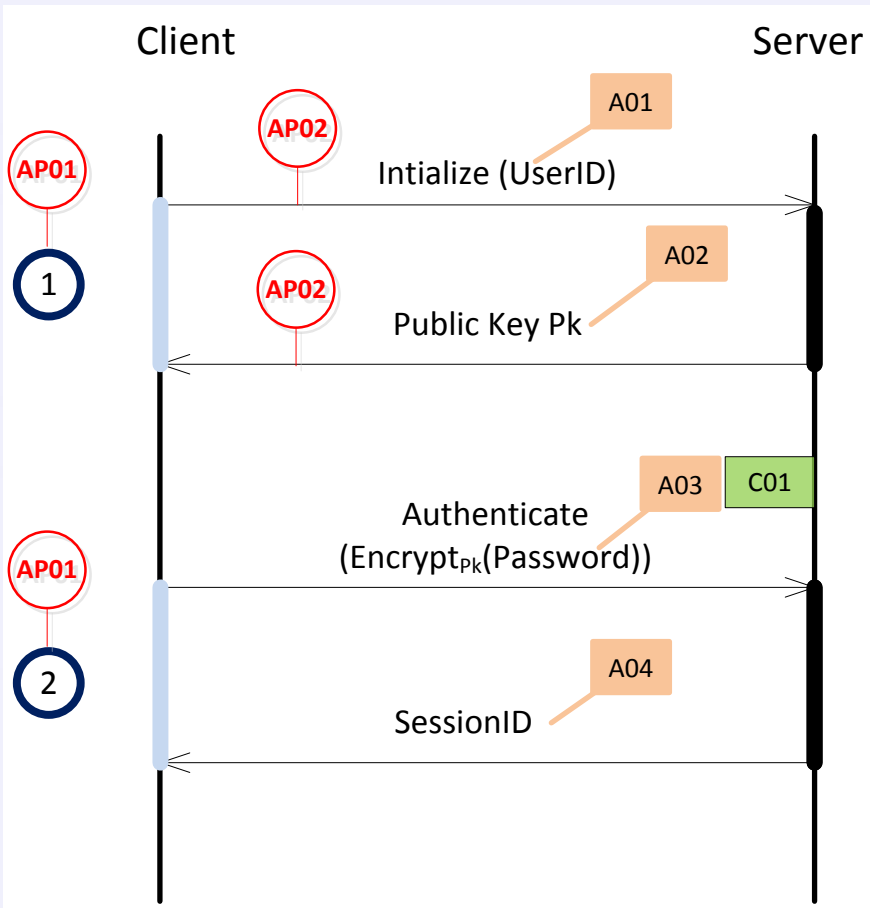
Assets

A01 – User ID
A02 – Public Key
A03 – Password
A04 – Session ID

Modeling the Protocol - Controls



Modeling the Protocol – Attacker Profile



Assets

A01 – User ID
A02 – Public Key
A03 – Password
A04 – Session ID

Controls

C01 – Encryption

Attacker Profiles

AP01 – Malicious Client
AP02 – Man in the Middle

Populate the Threat Table

OWASP
The Open Web Application Security Project

ID	Threat	Description	Attacker Profile	Asset	Control	Mitigation
1	Enumerate list of UserIDs	Call Initialize and interpret the response messages to determine valid UserIDs.	AP01	A01		Change the Initialize call to not take the UserID. Pass both the UserID and password in the same message and provide the same error message for all failures.
2	Hijack user session	AP02 eavesdrops on interaction 2, captures the SessionKey and hijacks the user's session.	AP02	A03		Encrypt the session ID using a shared, session-specific key.
3	Login as UserID by replaying the encrypted password	Eavesdrop on the interaction 2, capture the encrypted password, and then login as UserID replaying the encrypted password.	AP02	A04		Add a randomly generated nonce that's encrypted as part of the message.
4	Steal user passwords	Interpose between the client and server passing the client the attacker's private key.	AP02	A02		Send the public key in a CA-certified X.509 certificate.

OWASP
The Open Web Application Security Project

CONCLUDING REMARKS

It's a Wrap! (Tin)

OWASP
The Open Web Application Security Project

- Next steps at Intuit
  - Develop metrics to measure return on time investment
  - Scale the training for different locations, product teams
  - Have Product teams build the system diagrams
- Threat Model is valuable
  - Different risk levels have different security needs
  - Categorize projects into low, medium or high risk levels
  - There are other ways to secure a project
- Training is the key to success of the program
- Security team has to take the lead

It's a Wrap! (Scott)

OWASP
The Open Web Application Security Project

- Protocol Threat Models – a big win
  - Finer grained analysis even without code
  - Great for initialization sequences and any message flow needing crypto
- Threat modeling in partnership with development helps create the shared vision for proper remediation throughout the SDLC
- Build threat models around existing development artifacts rather than re-creating the wheel

OWASP
The Open Web Application Security Project

QUESTIONS

OWASP
The Open Web Application Security Project

THANK YOU FOR YOUR TIME