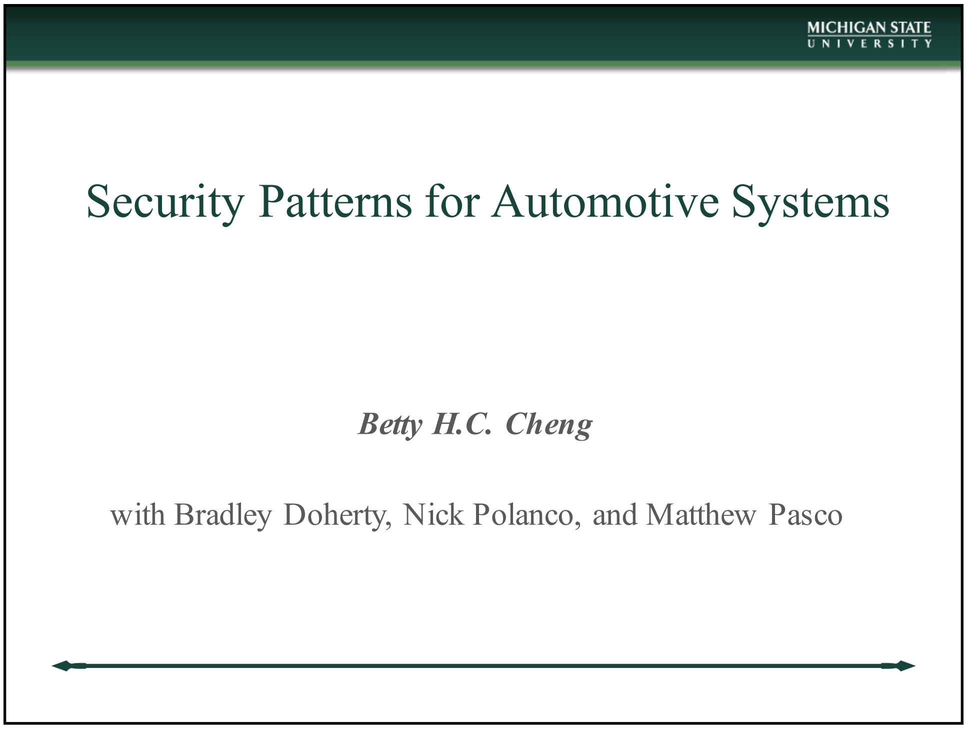
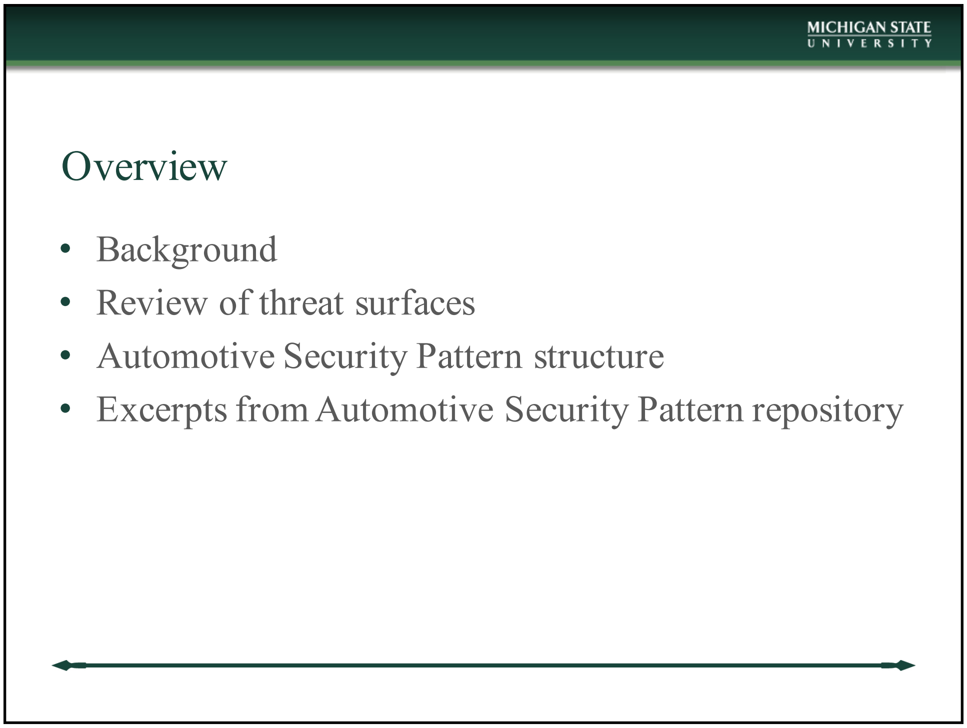


Security Patterns for Automotive Systems

Betty H.C. Cheng

with Bradley Doherty, Nick Polanco, and Matthew Pasco



Overview

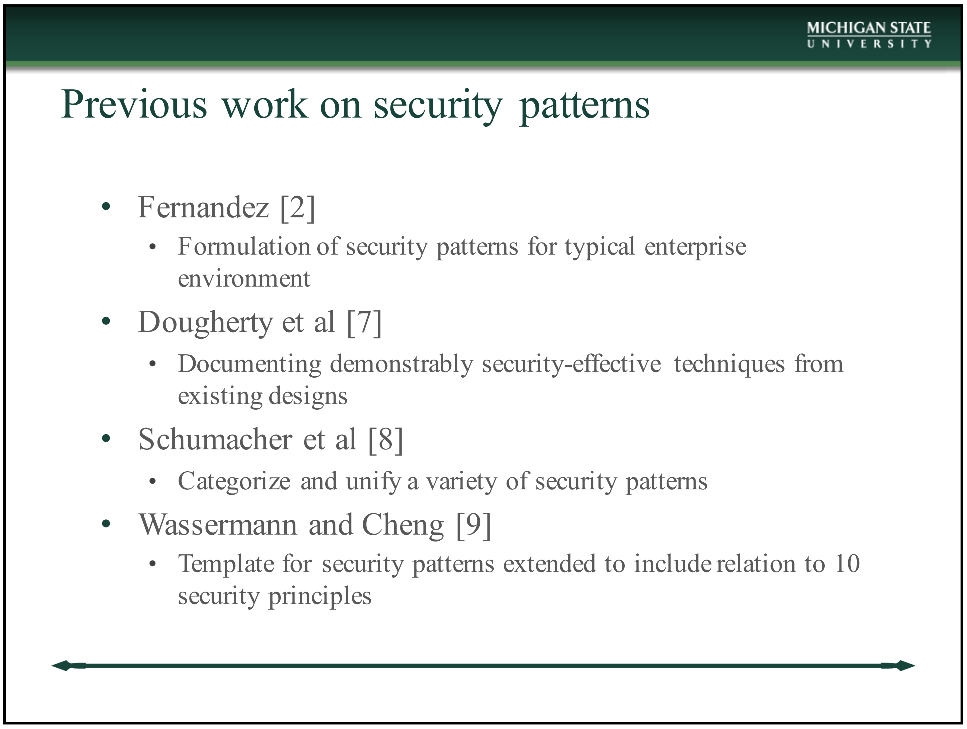
- Background
- Review of threat surfaces
- Automotive Security Pattern structure
- Excerpts from Automotive Security Pattern repository

Software Design Patterns

- Reuse of successful system designs
- Known solution to common problems
- Gamma et al. formulation: [1]
  - Pattern name
  - Problem addressed
  - Solution
  - Consequences of pattern use

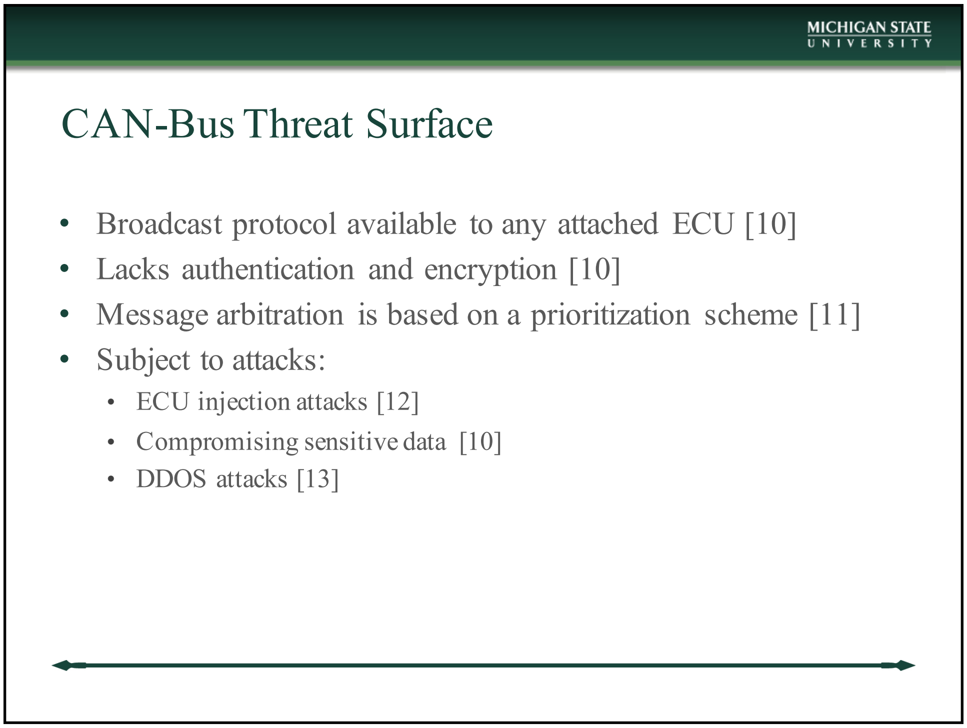
Security Patterns

- Used to manage threats to a given system [2]
- Security Patterns research active in several domains:
  - Distributed Systems [3]
  - Enterprise Systems [4]
  - Cloud Computing Systems [5]
- Security patterns can be applied to requirements gathering, design and implementation [6]



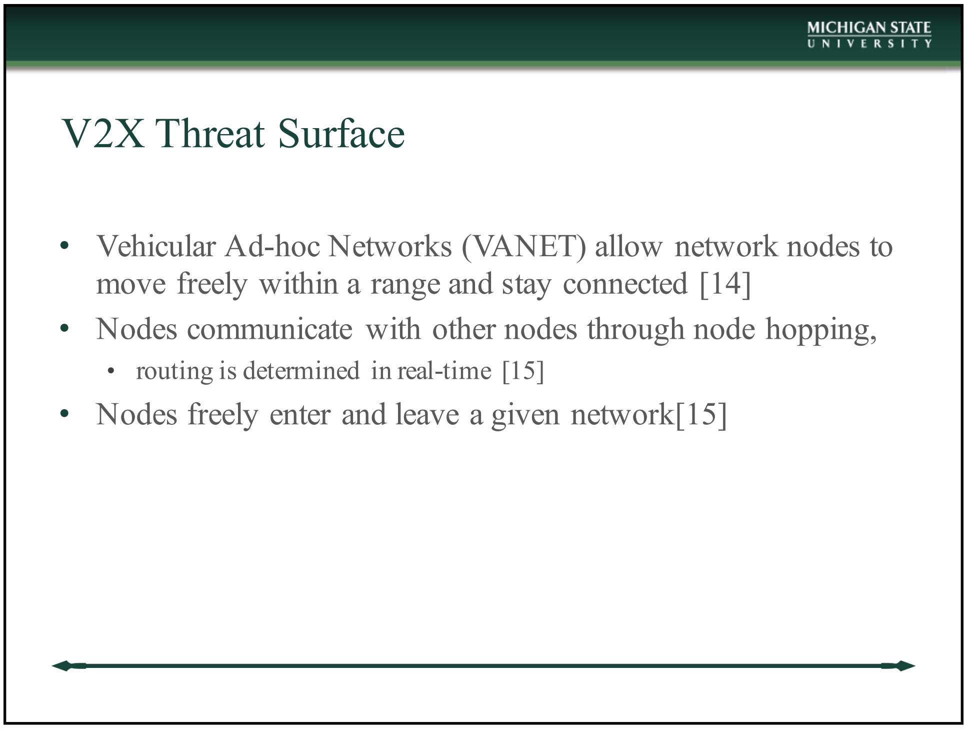
Previous work on security patterns

- Fernandez [2]
  - Formulation of security patterns for typical enterprise environment
- Dougherty et al [7]
  - Documenting demonstrably security-effective techniques from existing designs
- Schumacher et al [8]
  - Categorize and unify a variety of security patterns
- Wassermann and Cheng [9]
  - Template for security patterns extended to include relation to 10 security principles



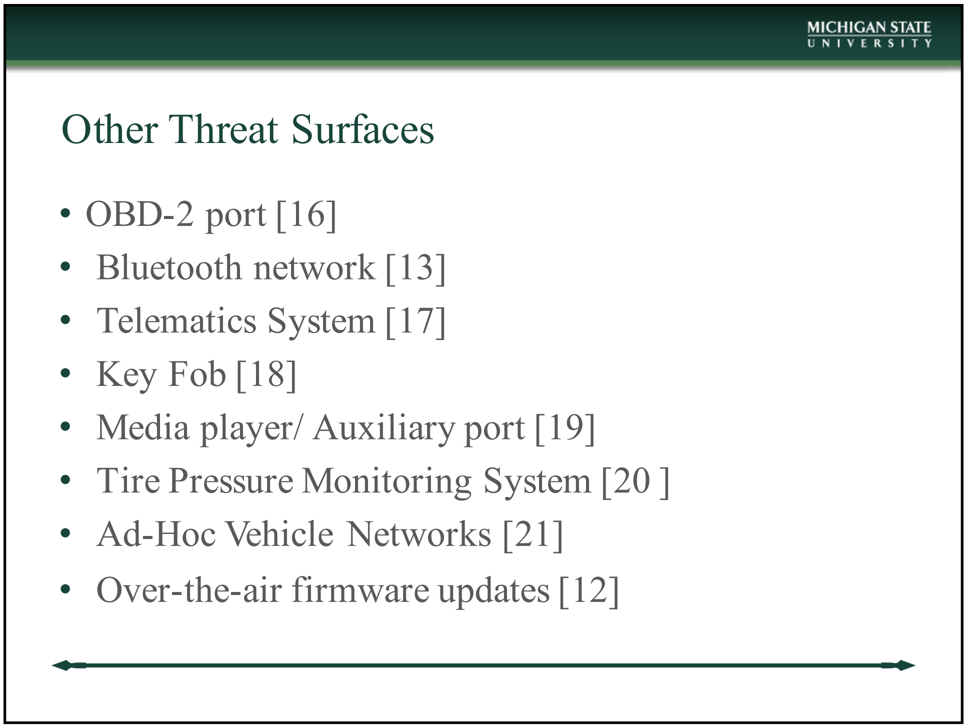
CAN-Bus Threat Surface

- Broadcast protocol available to any attached ECU [10]
- Lacks authentication and encryption [10]
- Message arbitration is based on a prioritization scheme [11]
- Subject to attacks:
  - ECU injection attacks [12]
  - Compromising sensitive data [10]
  - DDOS attacks [13]



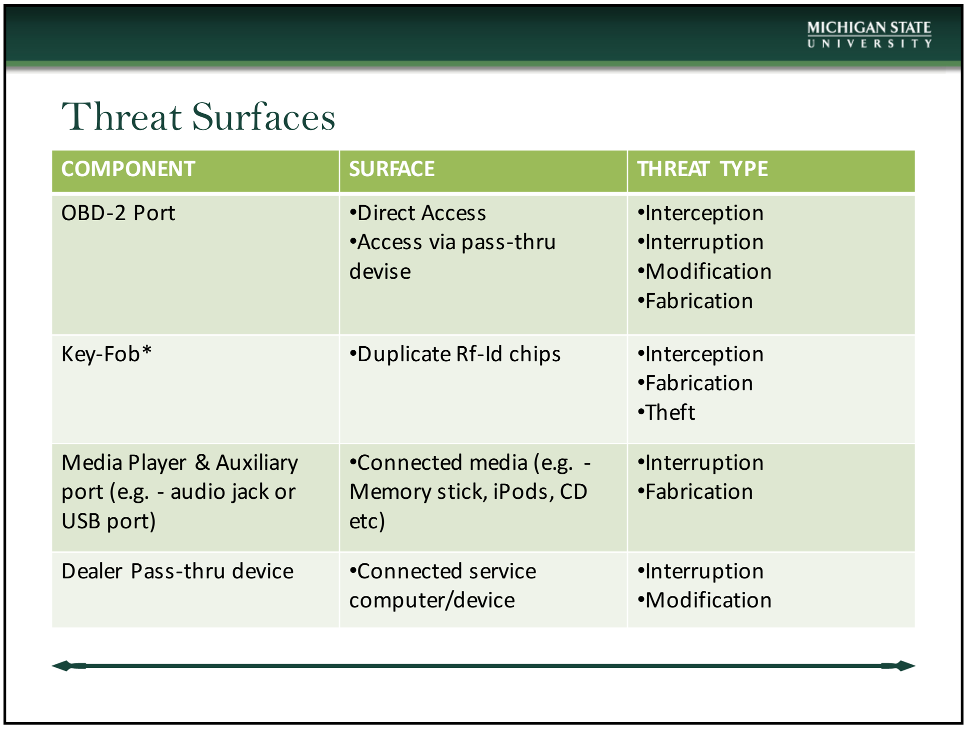
V2X Threat Surface

- Vehicular Ad-hoc Networks (VANET) allow network nodes to move freely within a range and stay connected [14]
- Nodes communicate with other nodes through node hopping,
  - routing is determined in real-time [15]
- Nodes freely enter and leave a given network[15]



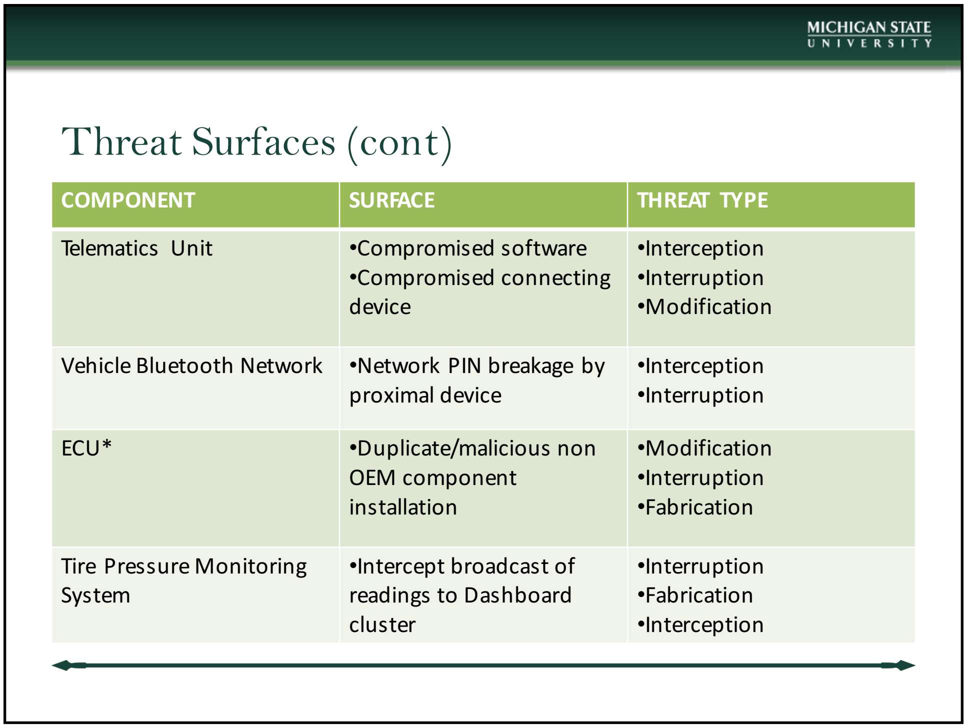
Other Threat Surfaces

- OBD-2 port [16]
- Bluetooth network [13]
- Telematics System [17]
- Key Fob [18]
- Media player/ Auxiliary port [19]
- Tire Pressure Monitoring System [20]
- Ad-Hoc Vehicle Networks [21]
- Over-the-air firmware updates [12]



Threat Surfaces

COMPONENT	SURFACE	THREAT TYPE
OBD-2 Port	•Direct Access •Access via pass-thru devise	•Interception •Interruption •Modification •Fabrication
Key-Fob*	•Duplicate Rf-Id chips	•Interception •Fabrication •Theft
Media Player & Auxiliary port (e.g. - audio jack or USB port)	•Connected media (e.g. - Memory stick, iPods, CD etc)	•Interruption •Fabrication
Dealer Pass-thru device	•Connected service computer/device	•Interruption •Modification



Threat Surfaces (cont)

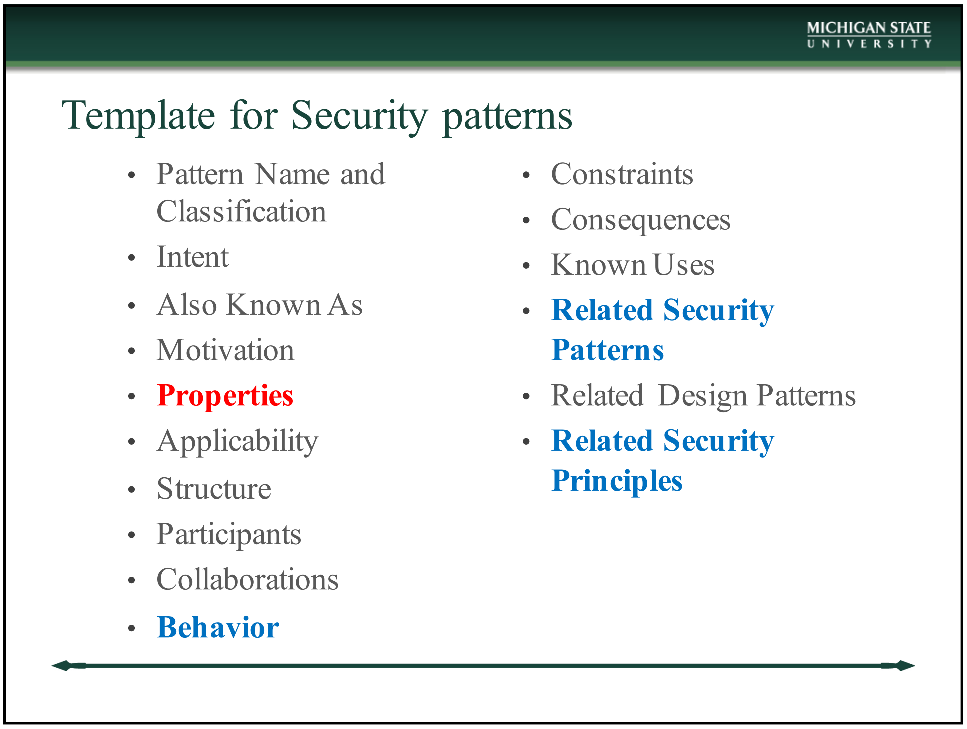
COMPONENT	SURFACE	THREAT TYPE
Telematics Unit	•Compromised software •Compromised connecting device	•Interception •Interruption •Modification
Vehicle Bluetooth Network	•Network PIN breakage by proximal device	•Interception •Interruption
ECU*	•Duplicate/malicious non OEM component installation	•Modification •Interruption •Fabrication
Tire Pressure Monitoring System	•Intercept broadcast of readings to Dashboard cluster	•Interruption •Fabrication •Interception

Threat Surfaces (cont)

COMPONENT	SURFACE	THREAT TYPE
Vehicular Ad-hoc Network	•Transmission from compromised node to another	•Interception •Interruption •Fabrication
Telematics Service	•Service parameters like I.P. address and subscriber identity module (if present)	•Interception •Interruption
Digital Car Radio	•Broadcast data processing	•Fabrication •Interruption

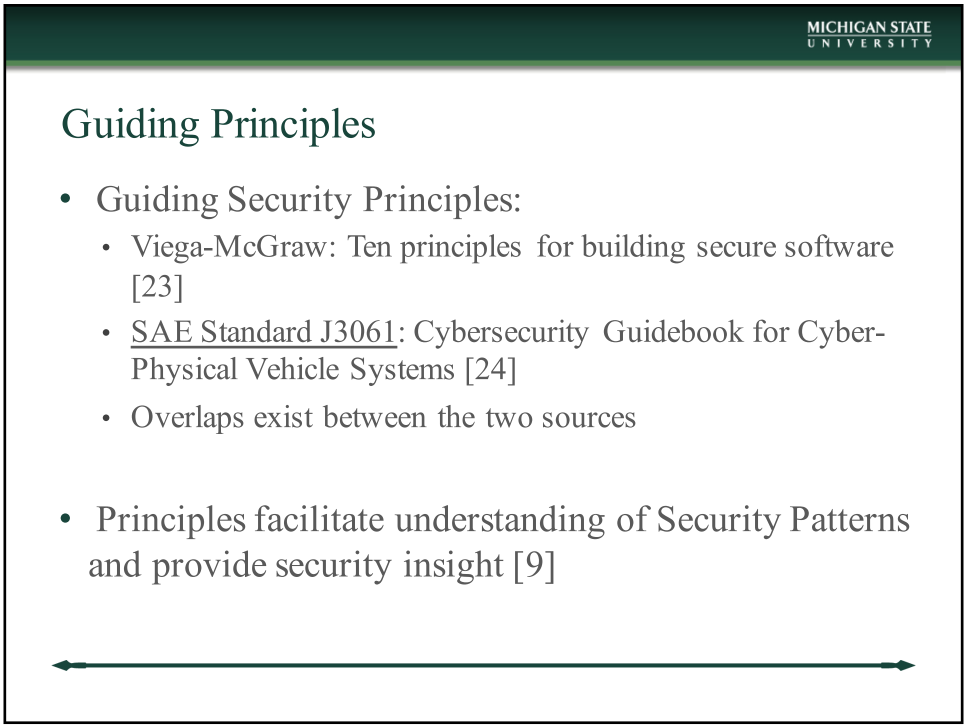
Template for Security Patterns

- Several templates have been used in previous security pattern research:
  - Security Patterns in Practice [2]
  - Security Patterns Repository [22]
  - Security Patterns: Technical Report [9]
- We constructed our template following the one defined by Gamma et al for general design patterns and extended by Wasserman and Cheng [9] for security-specific patterns
  - Incorporation of UML
  - Incorporation of guiding security principles



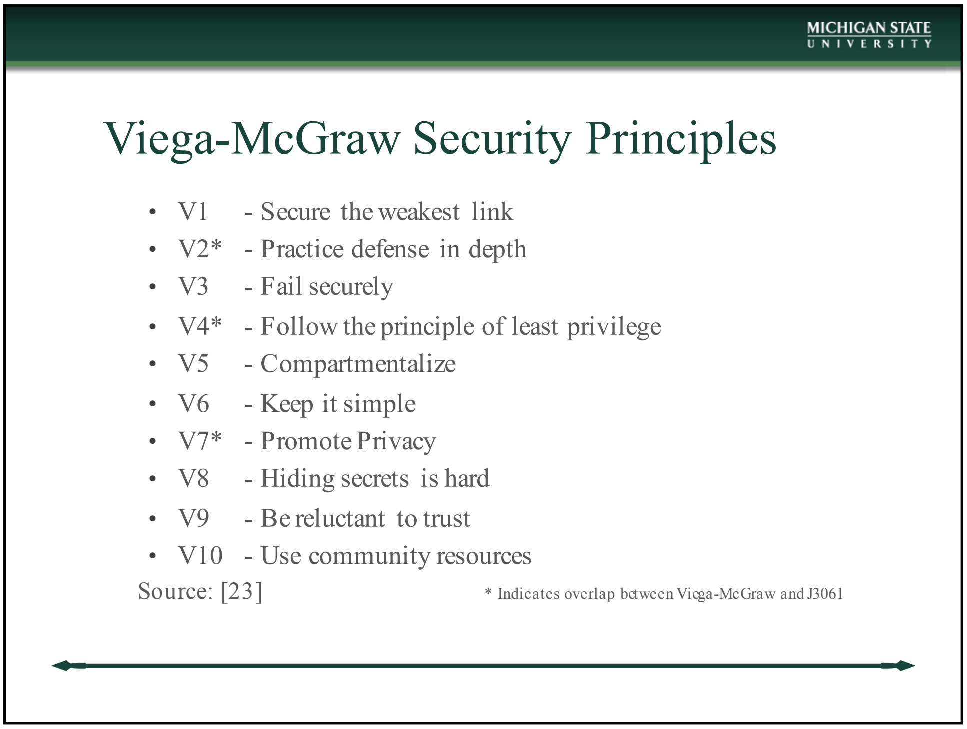
Template for Security patterns

- Pattern Name and Classification
- Intent
- Also Known As
- Motivation
- **Properties**
- Applicability
- Structure
- Participants
- Collaborations
- **Behavior**
- Constraints
- Consequences
- Known Uses
- **Related Security Patterns**
- Related Design Patterns
- **Related Security Principles**



Guiding Principles

- Guiding Security Principles:
  - Viega-McGraw: Ten principles for building secure software [23]
  - SAE Standard J3061: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems [24]
  - Overlaps exist between the two sources
- Principles facilitate understanding of Security Patterns and provide security insight [9]

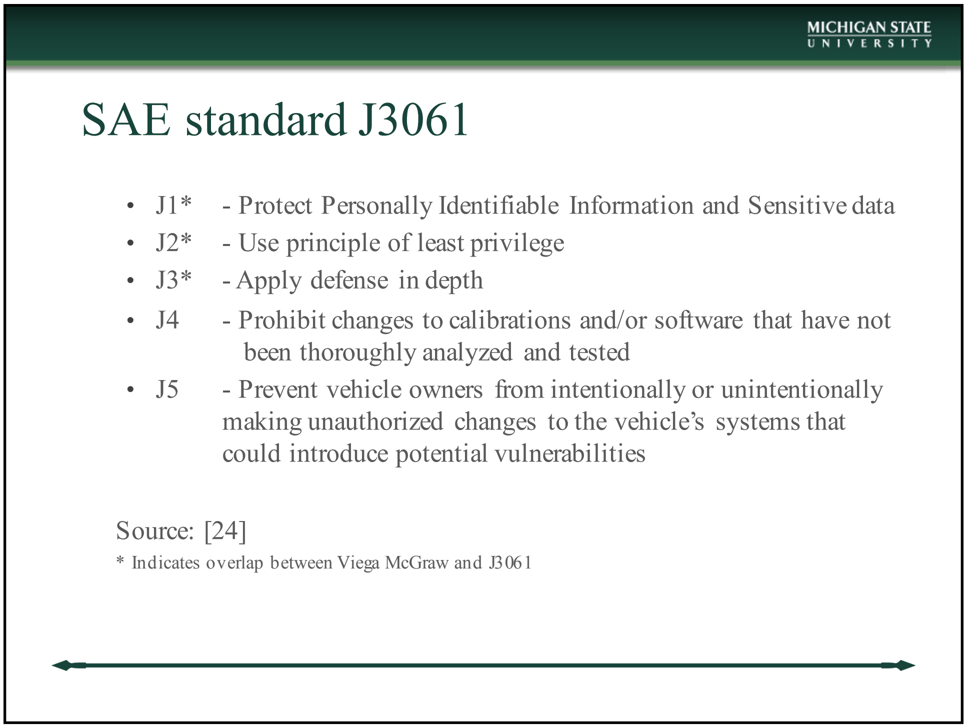


Viega-McGraw Security Principles

- V1 - Secure the weakest link
- V2* - Practice defense in depth
- V3 - Fail securely
- V4* - Follow the principle of least privilege
- V5 - Compartmentalize
- V6 - Keep it simple
- V7* - Promote Privacy
- V8 - Hiding secrets is hard
- V9 - Be reluctant to trust
- V10 - Use community resources

Source: [23]

* Indicates overlap between Viega-McGraw and J3061

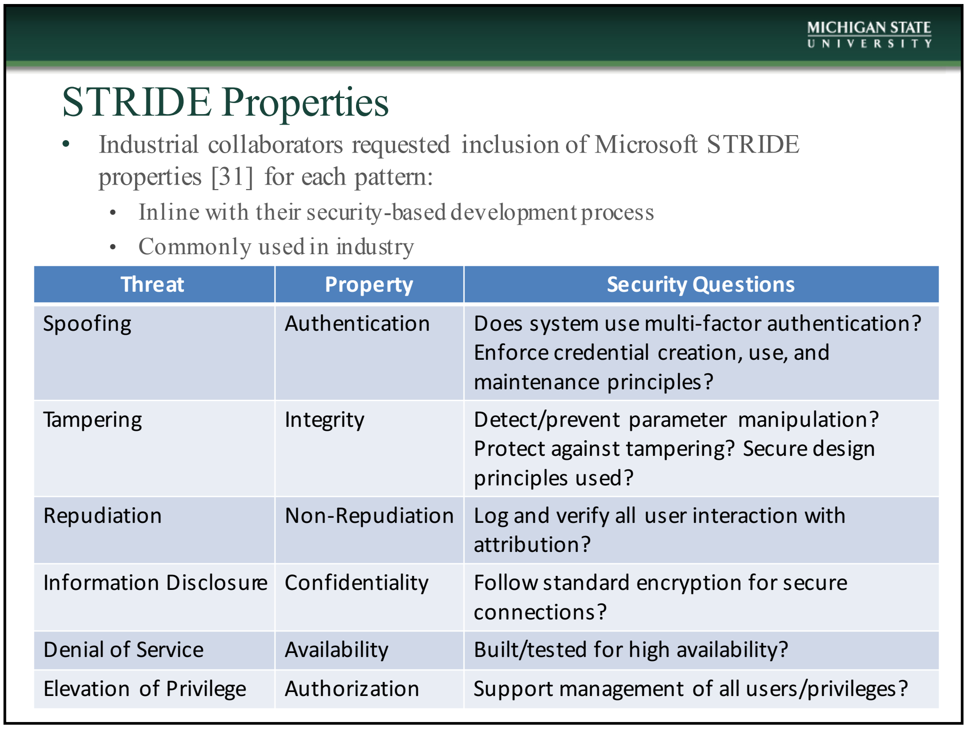


SAE standard J3061

- J1* - Protect Personally Identifiable Information and Sensitive data
- J2* - Use principle of least privilege
- J3* - Apply defense in depth
- J4 - Prohibit changes to calibrations and/or software that have not been thoroughly analyzed and tested
- J5 - Prevent vehicle owners from intentionally or unintentionally making unauthorized changes to the vehicle's systems that could introduce potential vulnerabilities

Source: [24]

* Indicates overlap between Viega McGraw and J3061

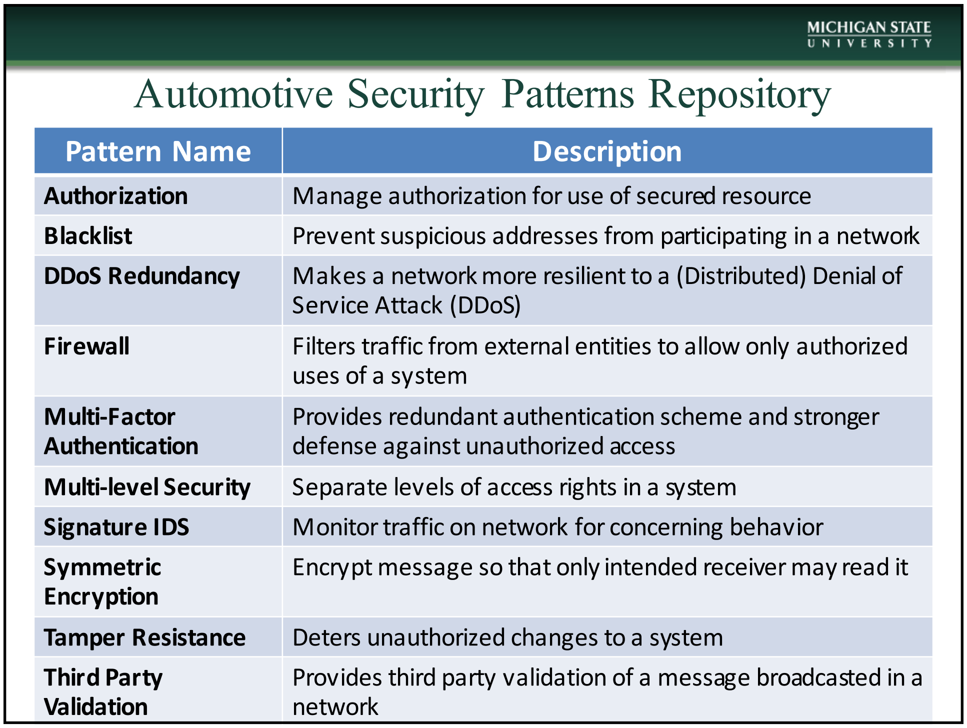


MICHIGAN STATE UNIVERSITY

STRIDE Properties

- Industrial collaborators requested inclusion of Microsoft STRIDE properties [31] for each pattern:
  - Inline with their security-based development process
  - Commonly used in industry

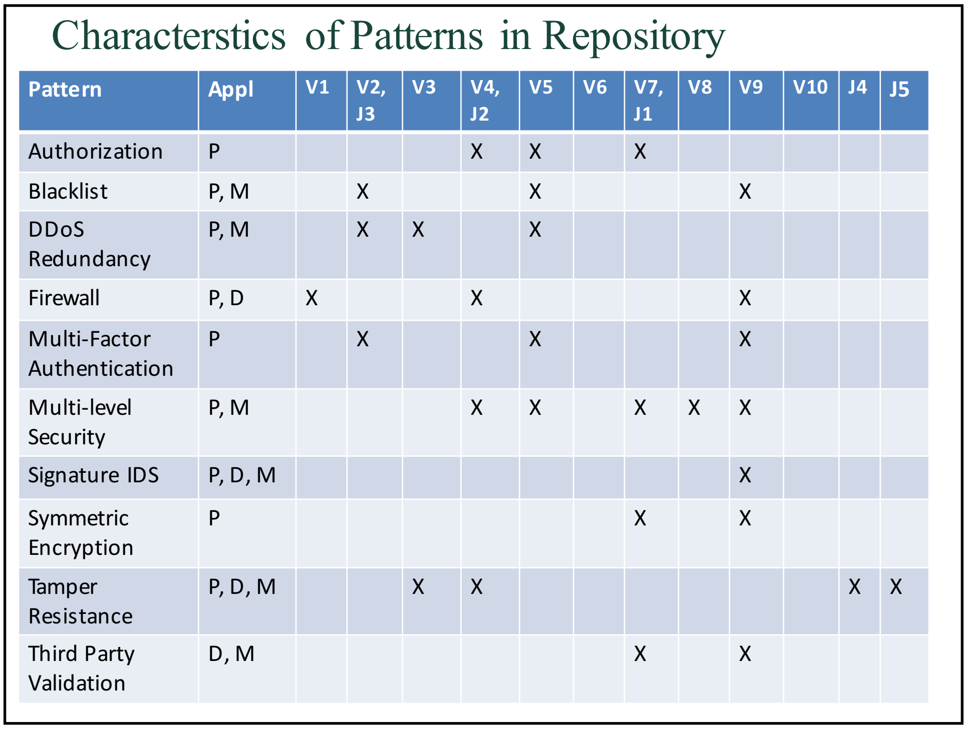
Threat	Property	Security Questions
Spoofing	Authentication	Does system use multi-factor authentication? Enforce credential creation, use, and maintenance principles?
Tampering	Integrity	Detect/prevent parameter manipulation? Protect against tampering? Secure design principles used?
Repudiation	Non-Repudiation	Log and verify all user interaction with attribution?
Information Disclosure	Confidentiality	Follow standard encryption for secure connections?
Denial of Service	Availability	Built/tested for high availability?
Elevation of Privilege	Authorization	Support management of all users/privileges?



MICHIGAN STATE UNIVERSITY

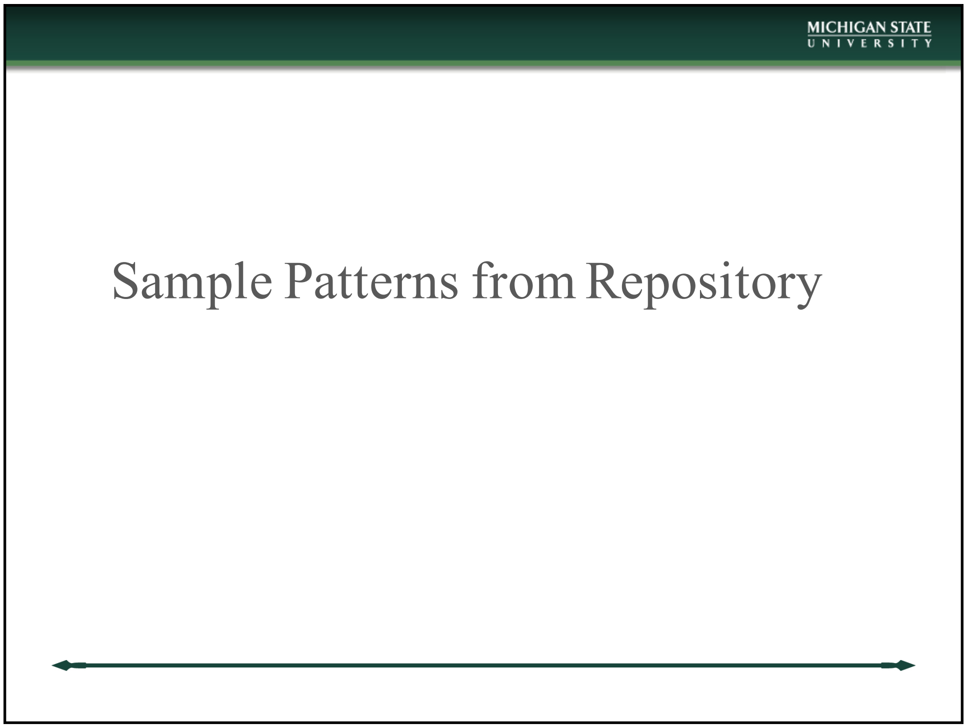
Automotive Security Patterns Repository

Pattern Name	Description
Authorization	Manage authorization for use of secured resource
Blacklist	Prevent suspicious addresses from participating in a network
DDoS Redundancy	Makes a network more resilient to a (Distributed) Denial of Service Attack (DDoS)
Firewall	Filters traffic from external entities to allow only authorized uses of a system
Multi-Factor Authentication	Provides redundant authentication scheme and stronger defense against unauthorized access
Multi-level Security	Separate levels of access rights in a system
Signature IDS	Monitor traffic on network for concerning behavior
Symmetric Encryption	Encrypt message so that only intended receiver may read it
Tamper Resistance	Deters unauthorized changes to a system
Third Party Validation	Provides third party validation of a message broadcasted in a network

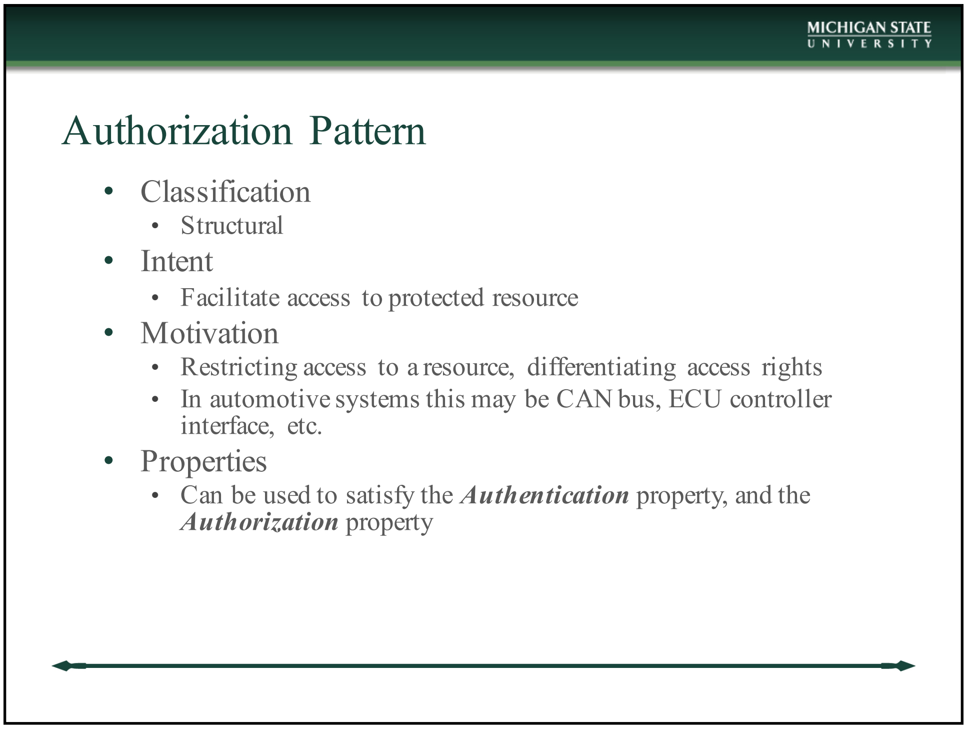


Characterstics of Patterns in Repository

Pattern	Appl	V1	V2, J3	V3	V4, J2	V5	V6	V7, J1	V8	V9	V10	J4	J5
Authorization	P				X	X		X					
Blacklist	P, M		X			X				X			
DDoS Redundancy	P, M		X	X		X							
Firewall	P, D	X			X					X			
Multi-Factor Authentication	P		X			X				X			
Multi-level Security	P, M				X	X		X	X	X			
Signature IDS	P, D, M									X			
Symmetric Encryption	P							X		X			
Tamper Resistance	P, D, M			X	X							X	X
Third Party Validation	D, M							X		X			



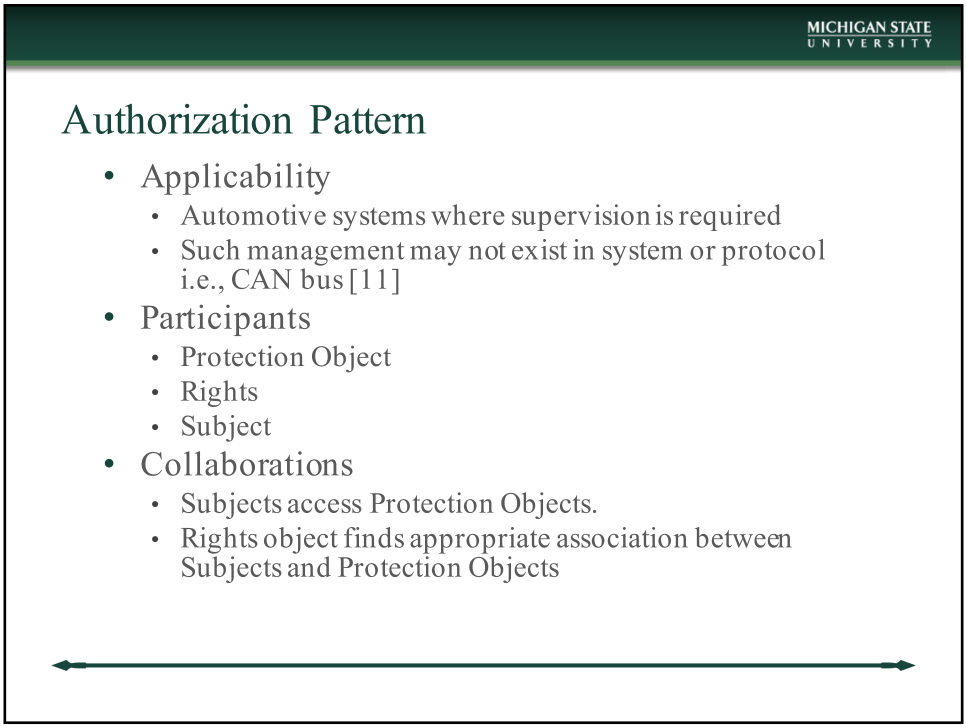
Sample Patterns from Repository



MICHIGAN STATE UNIVERSITY

Authorization Pattern

- Classification
  - Structural
- Intent
  - Facilitate access to protected resource
- Motivation
  - Restricting access to a resource, differentiating access rights
  - In automotive systems this may be CAN bus, ECU controller interface, etc.
- Properties
  - Can be used to satisfy the ***Authentication*** property, and the ***Authorization*** property



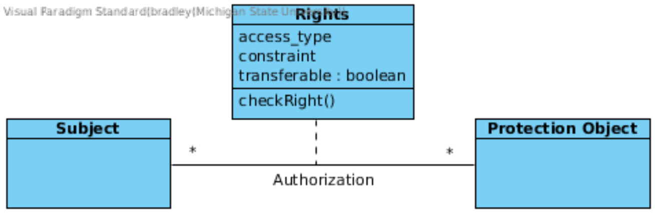
MICHIGAN STATE UNIVERSITY

Authorization Pattern

- Applicability
  - Automotive systems where supervision is required
  - Such management may not exist in system or protocol i.e., CAN bus [11]
- Participants
  - Protection Object
  - Rights
  - Subject
- Collaborations
  - Subjects access Protection Objects.
  - Rights object finds appropriate association between Subjects and Protection Objects

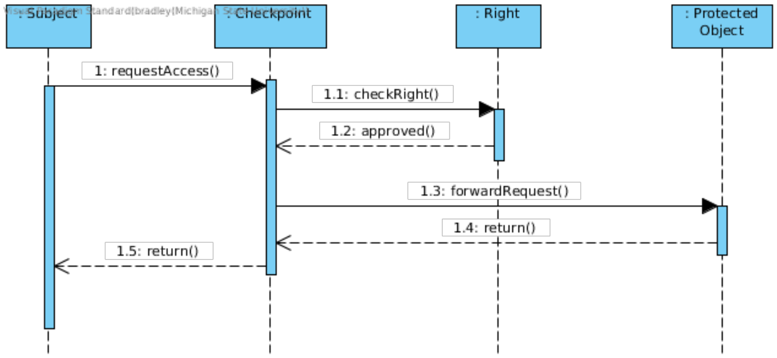
Authorization Pattern

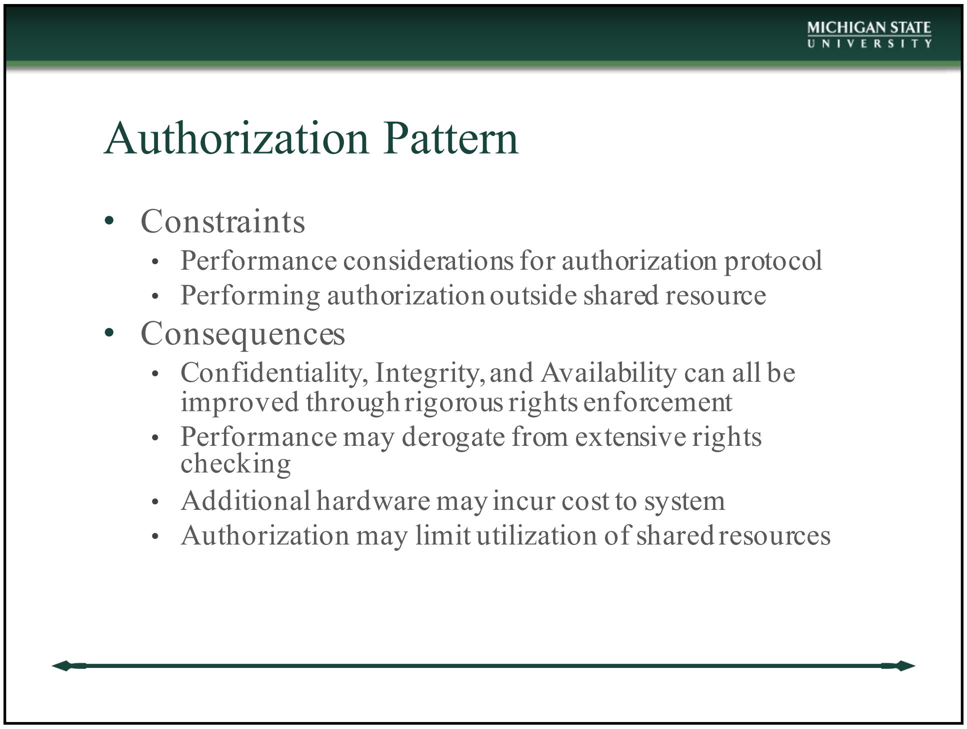
- Structure



Authorization Pattern

- Behavior

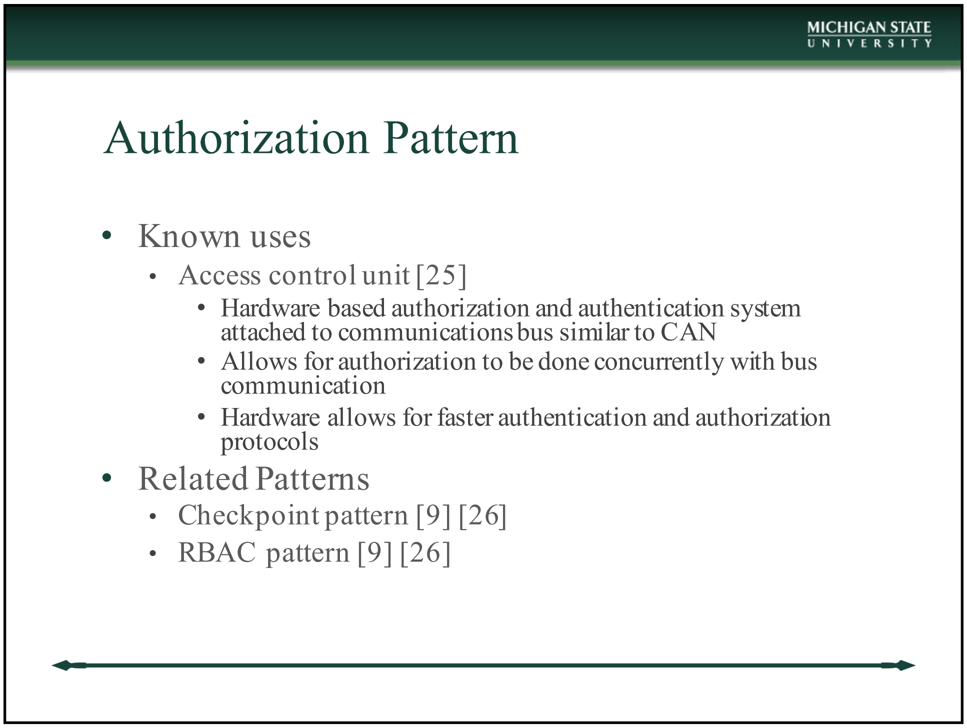




MICHIGAN STATE UNIVERSITY

Authorization Pattern

- Constraints
  - Performance considerations for authorization protocol
  - Performing authorization outside shared resource
- Consequences
  - Confidentiality, Integrity, and Availability can all be improved through rigorous rights enforcement
  - Performance may derogate from extensive rights checking
  - Additional hardware may incur cost to system
  - Authorization may limit utilization of shared resources



MICHIGAN STATE UNIVERSITY

Authorization Pattern

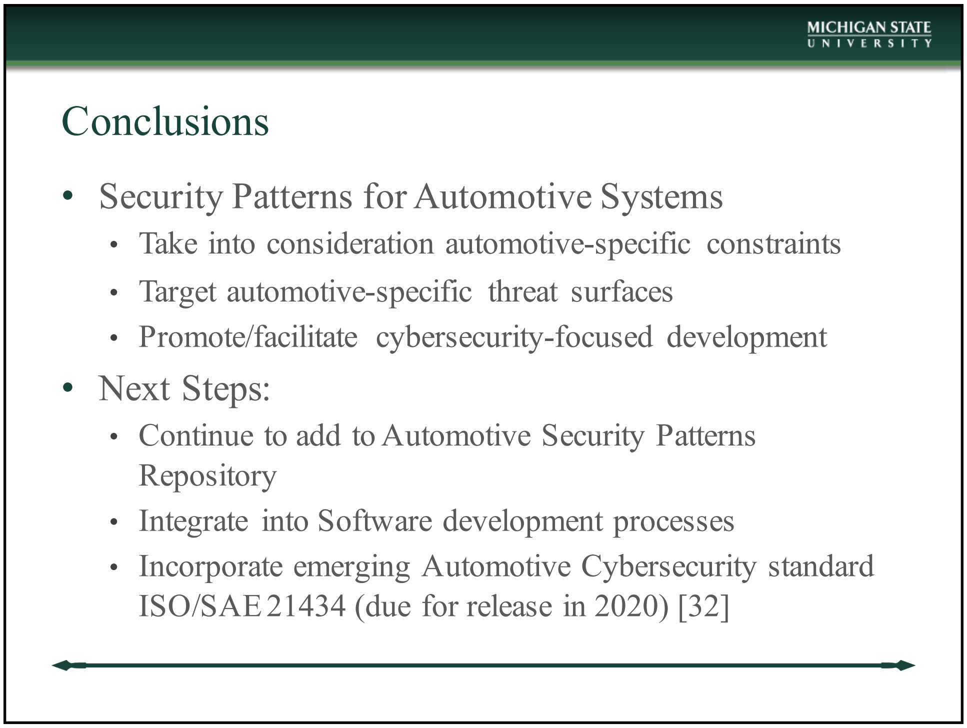
- Known uses
  - Access control unit [25]
    - Hardware based authorization and authentication system attached to communications bus similar to CAN
    - Allows for authorization to be done concurrently with bus communication
    - Hardware allows for faster authentication and authorization protocols
- Related Patterns
  - Checkpoint pattern [9] [26]
  - RBAC pattern [9] [26]

MICHIGAN STATE UNIVERSITY

Authorization Pattern

- Supported Principles
  - Least Privilege
  - Compartmentalization
  - Promotes Privacy

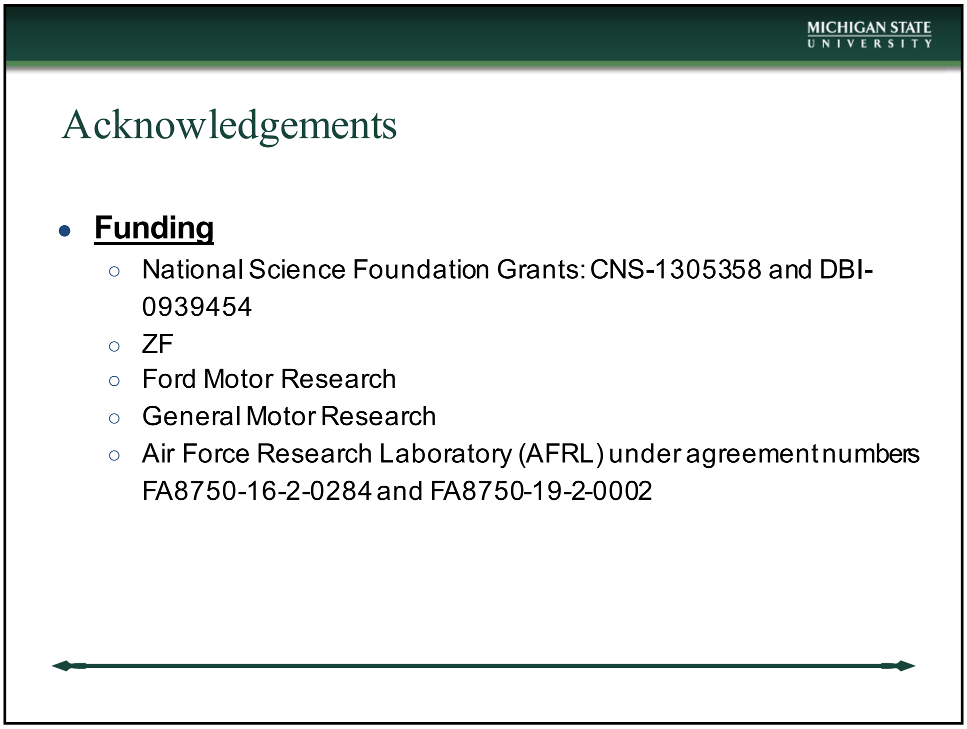
Skip to end



MICHIGAN STATE UNIVERSITY

Conclusions

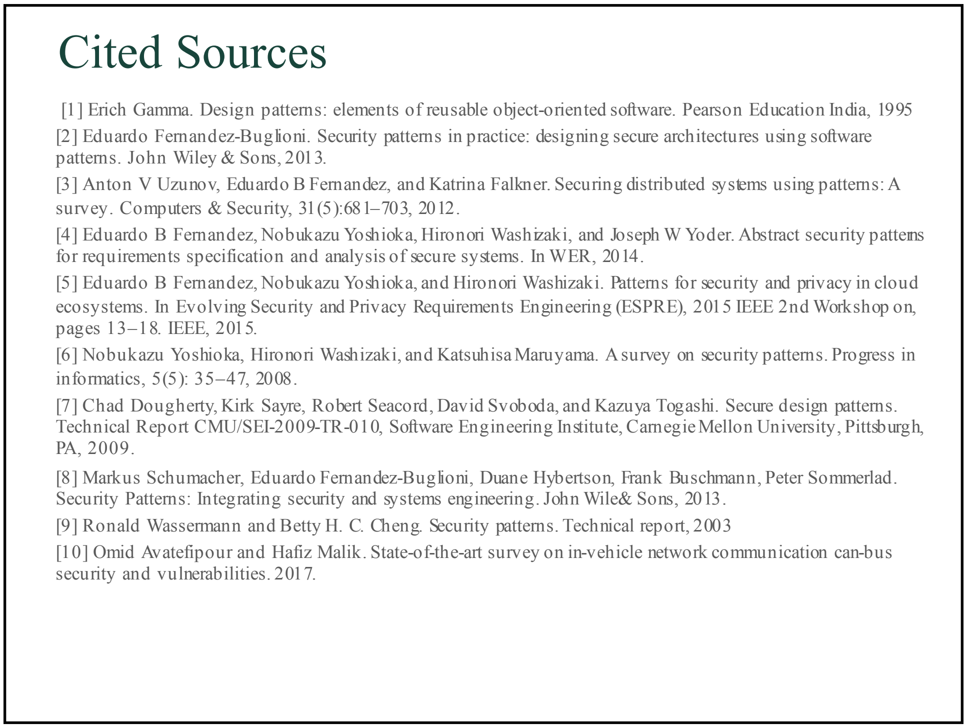
- Security Patterns for Automotive Systems
  - Take into consideration automotive-specific constraints
  - Target automotive-specific threat surfaces
  - Promote/facilitate cybersecurity-focused development
- Next Steps:
  - Continue to add to Automotive Security Patterns Repository
  - Integrate into Software development processes
  - Incorporate emerging Automotive Cybersecurity standard ISO/SAE 21434 (due for release in 2020) [32]



MICHIGAN STATE
UNIVERSITY

Acknowledgements

- **Funding**
  - National Science Foundation Grants: CNS-1305358 and DBI-0939454
  - ZF
  - Ford Motor Research
  - General Motor Research
  - Air Force Research Laboratory (AFRL) under agreement numbers FA8750-16-2-0284 and FA8750-19-2-0002



Cited Sources

[1] Erich Gamma. Design patterns: elements of reusable object-oriented software. Pearson Education India, 1995

[2] Eduardo Fernandez-Buglioni. Security patterns in practice: designing secure architectures using software patterns. John Wiley & Sons, 2013.

[3] Anton V Uzunov, Eduardo B Fernandez, and Katrina Falkner. Securing distributed systems using patterns: A survey. Computers & Security, 31(5):681–703, 2012.

[4] Eduardo B Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Joseph W Yoder. Abstract security patterns for requirements specification and analysis of secure systems. In WER, 2014.

[5] Eduardo B Fernandez, Nobukazu Yoshioka, and Hironori Washizaki. Patterns for security and privacy in cloud ecosystems. In Evolving Security and Privacy Requirements Engineering (ESPRE), 2015 IEEE 2nd Workshop on, pages 13–18. IEEE, 2015.

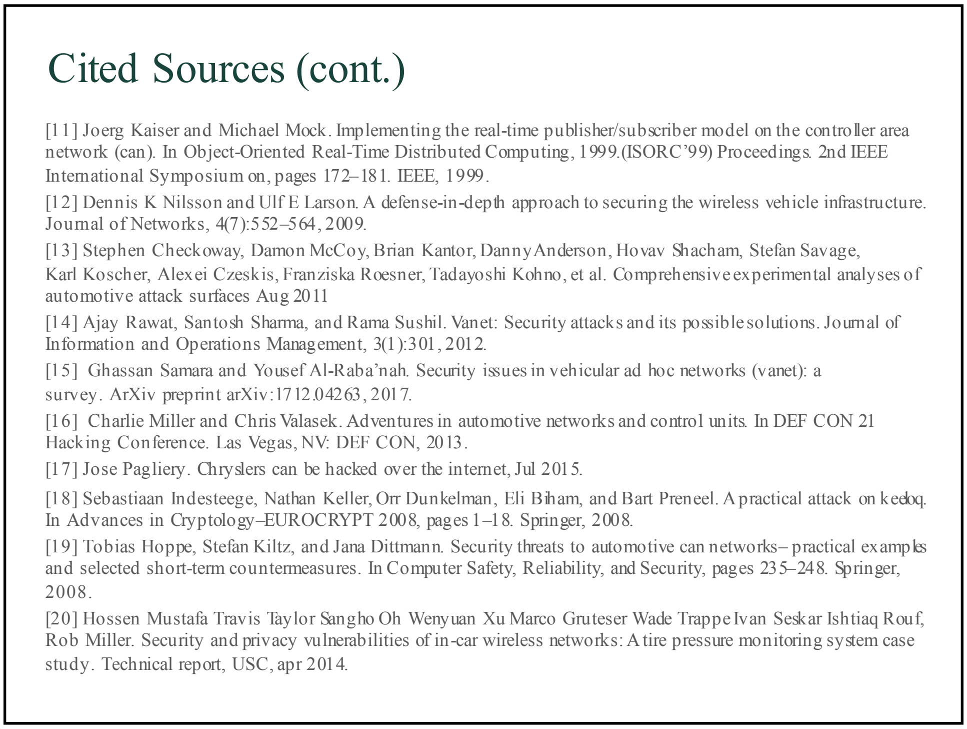
[6] Nobukazu Yoshioka, Hironori Washizaki, and Katsuhisa Maruyama. A survey on security patterns. Progress in informatics, 5(5): 35–47, 2008.

[7] Chad Dougherty, Kirk Sayre, Robert Seacord, David Svoboda, and Kazuya Togashi. Secure design patterns. Technical Report CMU/SEI-2009-TR-010, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2009.

[8] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, Peter Sommerlad. Security Patterns: Integrating security and systems engineering. John Wile& Sons, 2013.

[9] Ronald Wassermann and Betty H. C. Cheng. Security patterns. Technical report, 2003

[10] Omid Avatefipour and Hafiz Malik. State-of-the-art survey on in-vehicle network communication can-bus security and vulnerabilities. 2017.



Cited Sources (cont.)

[11] Joerg Kaiser and Michael Mock. Implementing the real-time publisher/subscriber model on the controller area network (can). In Object-Oriented Real-Time Distributed Computing, 1999.(ISORC'99) Proceedings. 2nd IEEE International Symposium on, pages 172–181. IEEE, 1999.

[12] Dennis K Nilsson and Ulf E Larson. A defense-in-depth approach to securing the wireless vehicle infrastructure. Journal of Networks, 4(7):552–564, 2009.

[13] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces Aug 2011

[14] Ajay Rawat, Santosh Sharma, and Rama Sushil. Vanet: Security attacks and its possible solutions. Journal of Information and Operations Management, 3(1):301, 2012.

[15] Ghassan Samara and Yousef Al-Raba'nah. Security issues in vehicular ad hoc networks (vanet): a survey. ArXiv preprint arXiv:1712.04263, 2017.

[16] Charlie Miller and Chris Valasek. Adventures in automotive networks and control units. In DEF CON 21 Hacking Conference. Las Vegas, NV: DEF CON, 2013.

[17] Jose Pagliery. Chryslers can be hacked over the internet, Jul 2015.

[18] Sebastiaan Indesteege, Nathan Keller, Orr Dunkelman, Eli Biham, and Bart Preneel. A practical attack on keeloq. In Advances in Cryptology–EUROCRYPT 2008, pages 1–18. Springer, 2008.

[19] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Security threats to automotive can networks– practical examples and selected short-term countermeasures. In Computer Safety, Reliability, and Security, pages 235–248. Springer, 2008.

[20] Hossen Mustafa Travis Taylor Sangho Oh Wenyuan Xu Marco Gruteser Wade Trappe Ivan Seskar Ishtiaq Rouf, Rob Miller. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. Technical report, USC, apr 2014.

Cited Sources

[21] RR Brooks, S Sander, Juan Deng, and Joachim Taiber. Automobile security concerns. Vehicular Technology Magazine, IEEE, 4(2):52–64, 2009.

[22] Darrell M Kienzle, Matthew C Elder, David Tyree, and James Edwards-Hewitt. Security patterns repository version 1.0. DARPA, Washington DC, 2002.

[23] John Viega and Gary McGraw. Building Secure Software - How to Avoid Security Problems the Right Way. Addison-Wesley, September 2002

[24] Vehicle Cybersecurity Systems Engineering Committee. J3061 cybersecurity guidebook for cyber-physical vehicle systems. Technical report, SAE International, 2016.

[25] Mankuan M Vai, Roger I Khazan, Daniil M Utin, Sean R O'Melia, David J Whelihan, and Benjamin R Nahill. Secure embedded systems. Technical report, MIT Lincoln Laboratory Lexington United States, 2016

[26] J. Yoder and J. Barcalow. Architectural patterns for enabling application security, 1997

[27] Qiyan Wang and Sanjay Sawhney. Vecure: A practical security framework to protect the can bus of vehicles. In Internet of Things (IOT), 2014 International Conference on the, pages 13–18. IEEE, 2014

[28] Kyong-Tak Cho and Kang G Shin. Fingerprinting electronic control units for vehicle intrusion detection. In USENIX Security Symposium, pages 911–927, 2016.

[29] Syed Rizvi, Jonathan Willet, Donte Perino, Seth Marasco, and Chandler Condo. A threat to vehicular cyber security and the urgency for correction. Procedia Computer Science, 114:100– 105, 2017.

[30] Ameneh Daeinabi and Akbar Ghaffarpour Rahbar. Detection of malicious vehicles (dmv) through monitoring in vehicular ad-hoc networks. Multimedia tools and applications, 66(2):325–338, 2013.

[31] Microsoft Corporation,"Security development lifecycle," June 2018.

[32] C. Schmittner, G .Griessnig, and Z. Ma, "Status of the development of ISO/SAE 21434," in *25th European Conference, EuroSPI 2018, Bilbao, Spain, September 5-7, 2018, Proceedings*, pp. 504–513, EuroSPI, 01 2018.