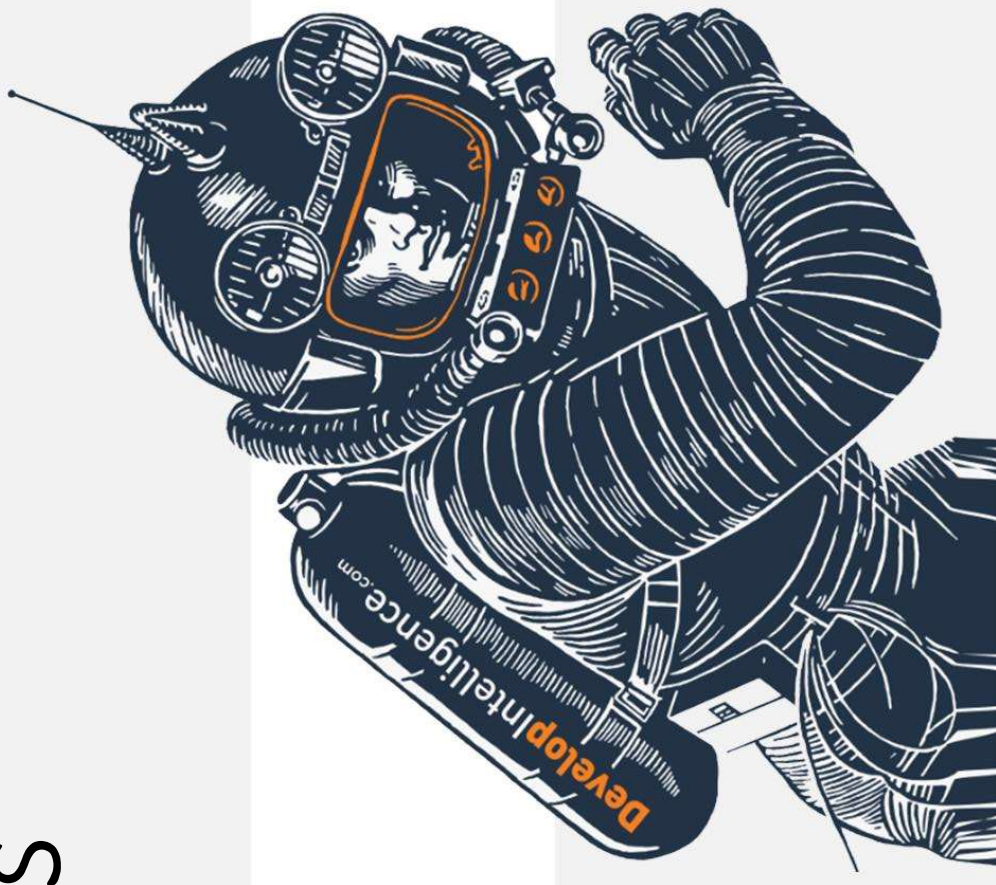


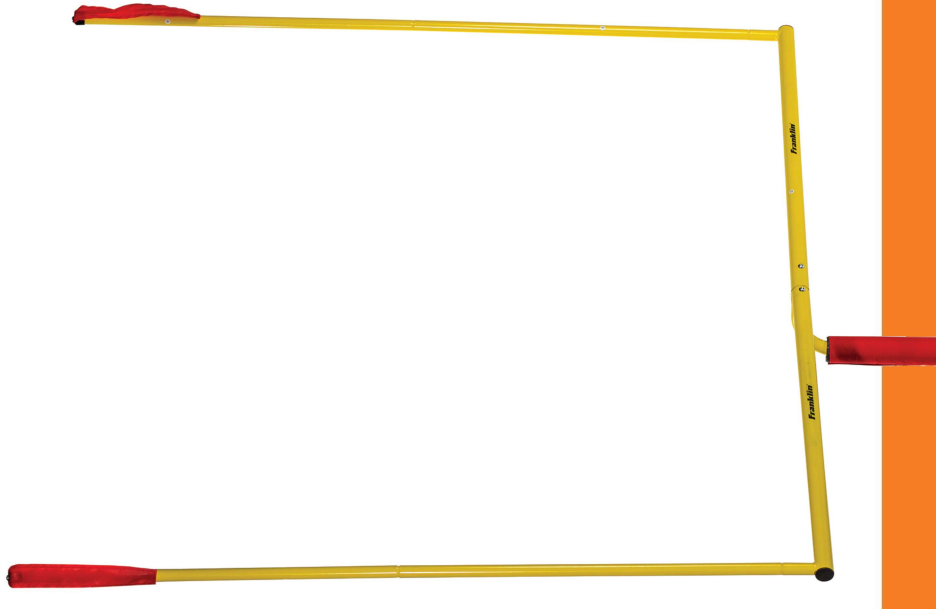
Guiding Principles





Goals

- Discuss the merits of security theater
- List 2 of the basicest rules





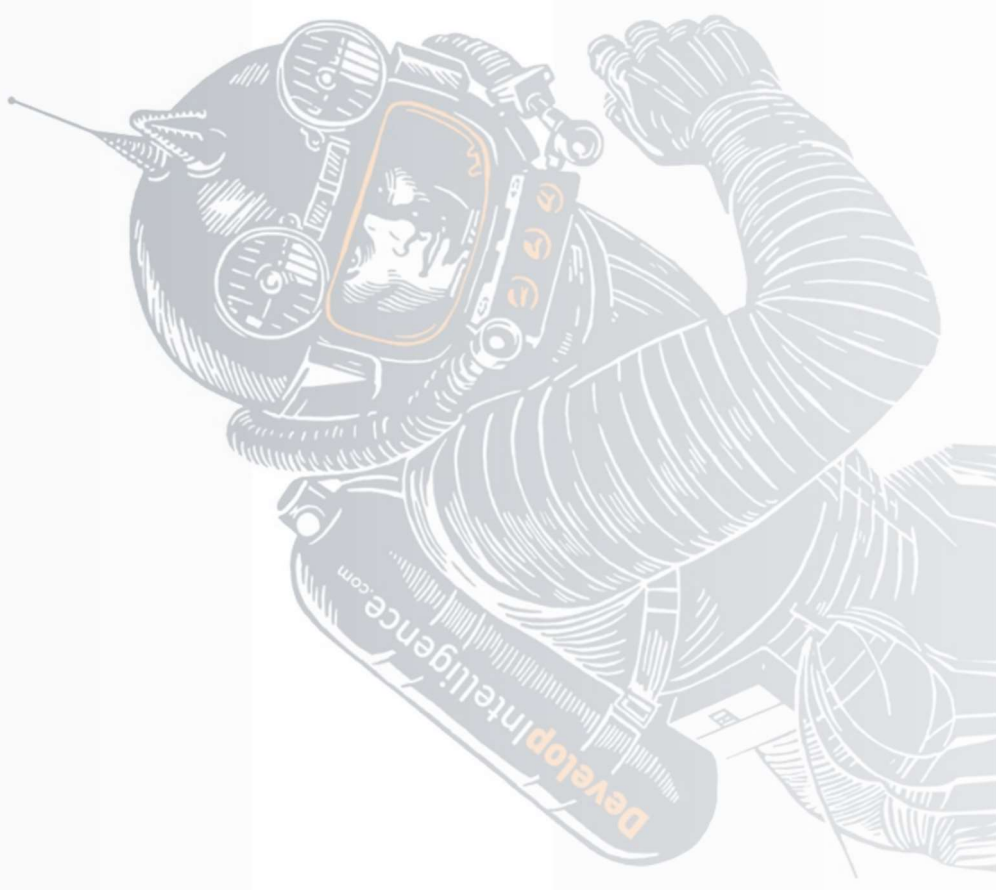
Roadmap



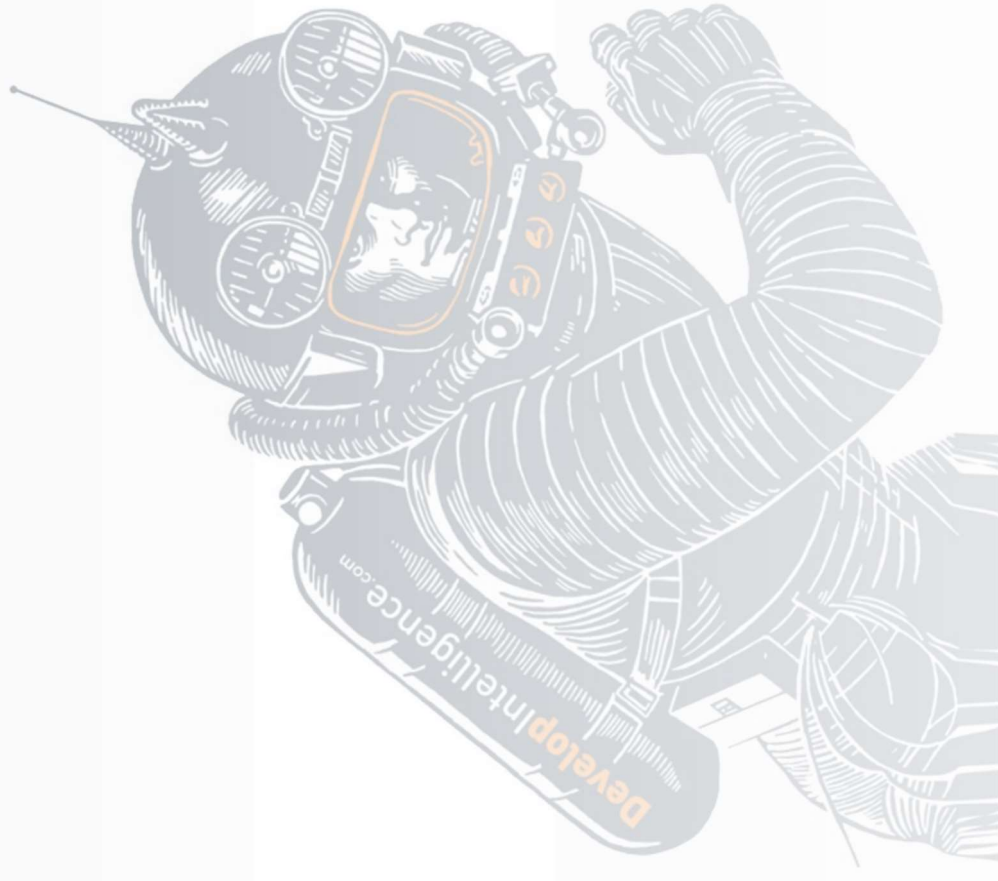
Develop
Intelligence

1. Scary Stories
2. Mindset
3. Basicist Rules for Web Apps





Scary Stories





Formative Experience

- Mentor
- Adorably, unintentionally problematic
- Lessons:
 - Red team
 - Beginner's mind





Biggest Data Breaches

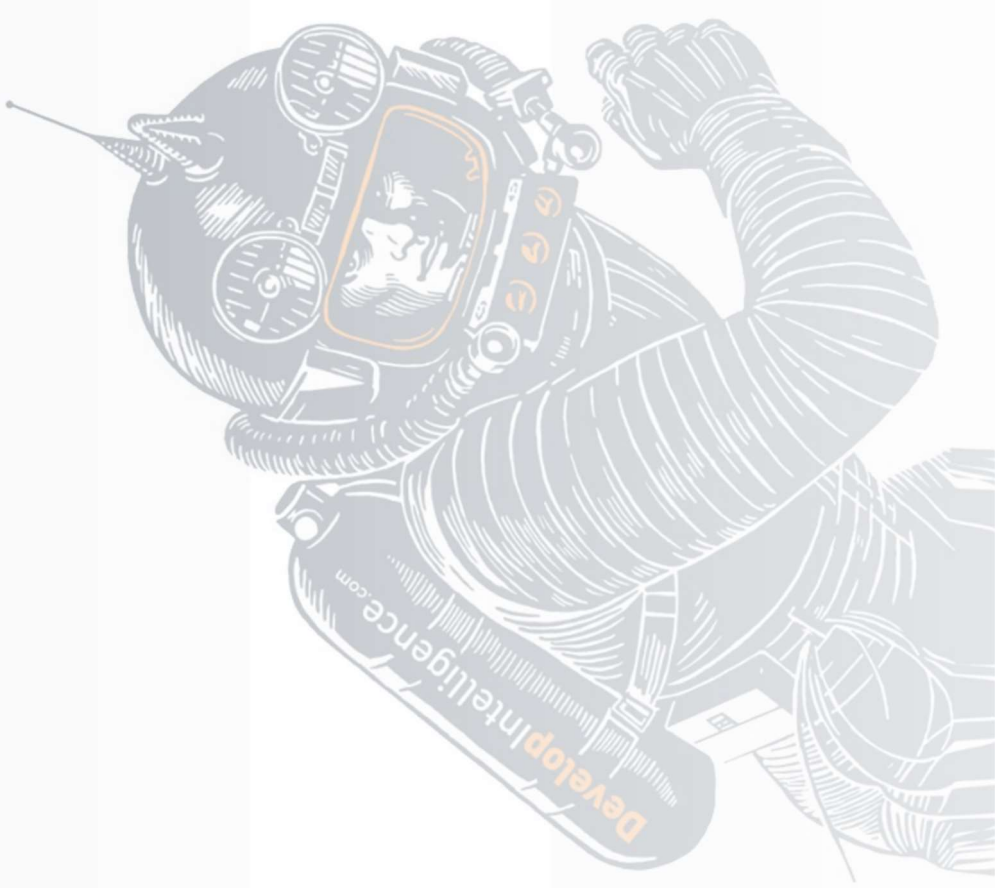
Victim	Year	Users Affected (M)
Adobe	2013	152
7-Eleven	2013	160
Marriott	2018	500
Facebook	2019	540
First American	2019	885



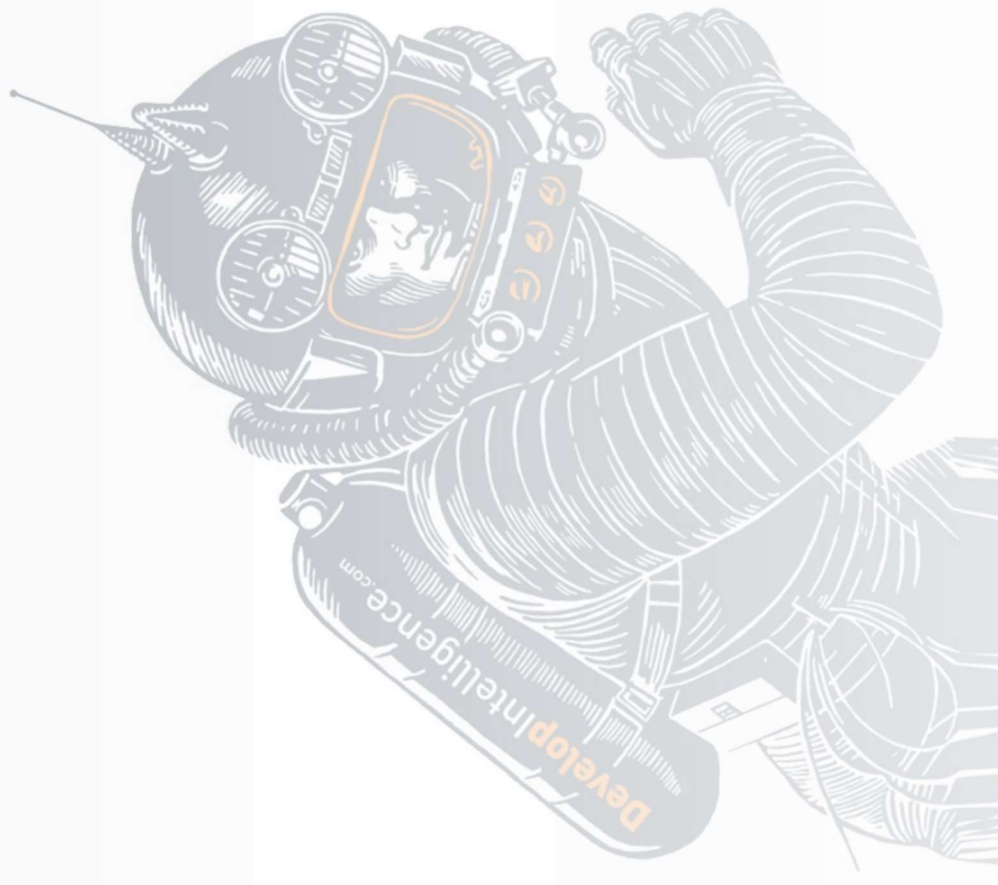
Worldwide Cybercrime Damages



- Cost \$3 trillion in 2015
- \$6 trillion in 2021
- US Federal Revenue in 2019: \$3.46 trillion



Mindset





Overview

1. Not Binary
2. Relative
3. Process



Security is Not Binary

The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.





Security is relative

- There's no door that the NSA can't get in, but that doesn't matter.
- (Hereafter referred to as a skilled and determined hacker.)
- But even with those resources, it's not worth the money.



Security is a Feature

- It's just for a long time, no one was very articulate
- They just say, it should 'be secure'
- Think about your risk profile and risk tolerance



Alignment is Hard

Agreement is easy... alignment is hard. When I ask my wife if she wants dinner, she quickly says okay-- the trouble starts when I ask where we should go. -- Mark Magliocco



Security Should be Redundant

- Like a Motte-and-bailey castle
- Example:
 - Secure your identity database
 - Hash and salt passwords



Stand on the shoulders of giants

- Don't roll your own stuff
- Think hard before adopting non-mainstream stuff (e.g. Choose Vue over Elm (or whatever))
- Avoid dependencies. If you don't need dancing bunnies, or whatever, maybe roll your own.
- Follow best-practices



Bear Joke

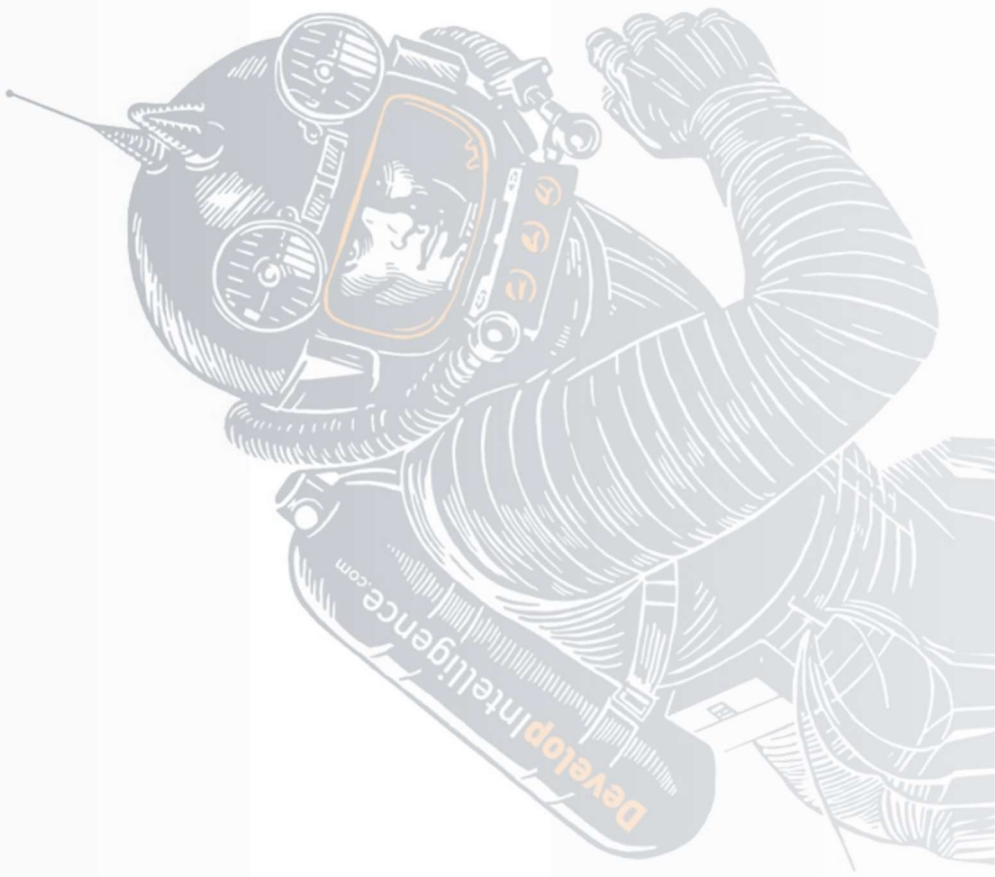




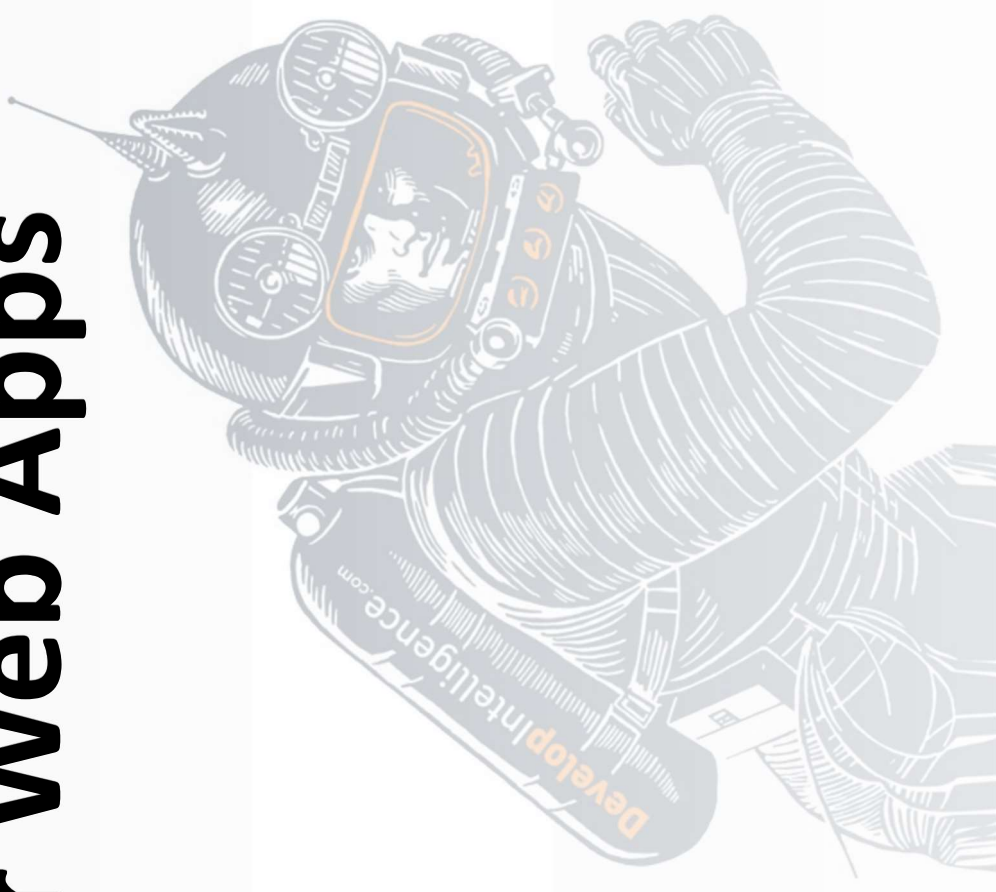
Think Like a Practitioner

- The one time I got fired
- Hippocratic oath
- Responsibility to users





Basicist Rules for Web Apps





Think about **Trust**

- Can you trust the integrity an HTTP request?
- Answer: No
- What about internally-facing services?
- Also: No



Validate Input

*Input **validation** is the process of ensuring input data is consistent with application expectations. Data that falls outside of an expected set of values can cause our application to yield unexpected results, for example violating business logic, triggering faults, and even allowing an attacker to take control of resources or the application itself.*



Never Trust Input

- When building a client, assume the server is compromised
- Building an API, assume you're talking to a hacker
- Validate Everything



Bad Input Validation Enables

- Cross Site Scripting
- SQL Injection





Encode HTML

- HTML is a very, very permissive format
- Browsers try their best to render the content, even if it is malformed



Bind Parameters for Database



Queries

- **Don't** concatenate strings
- Use a prepared statement e.g. SqlCommand
- Better yet EF Core



Protect Data in Transit



- SSL Everywhere





Follow Checklists

- Look it up
- cheatsheetseries.owasp.org



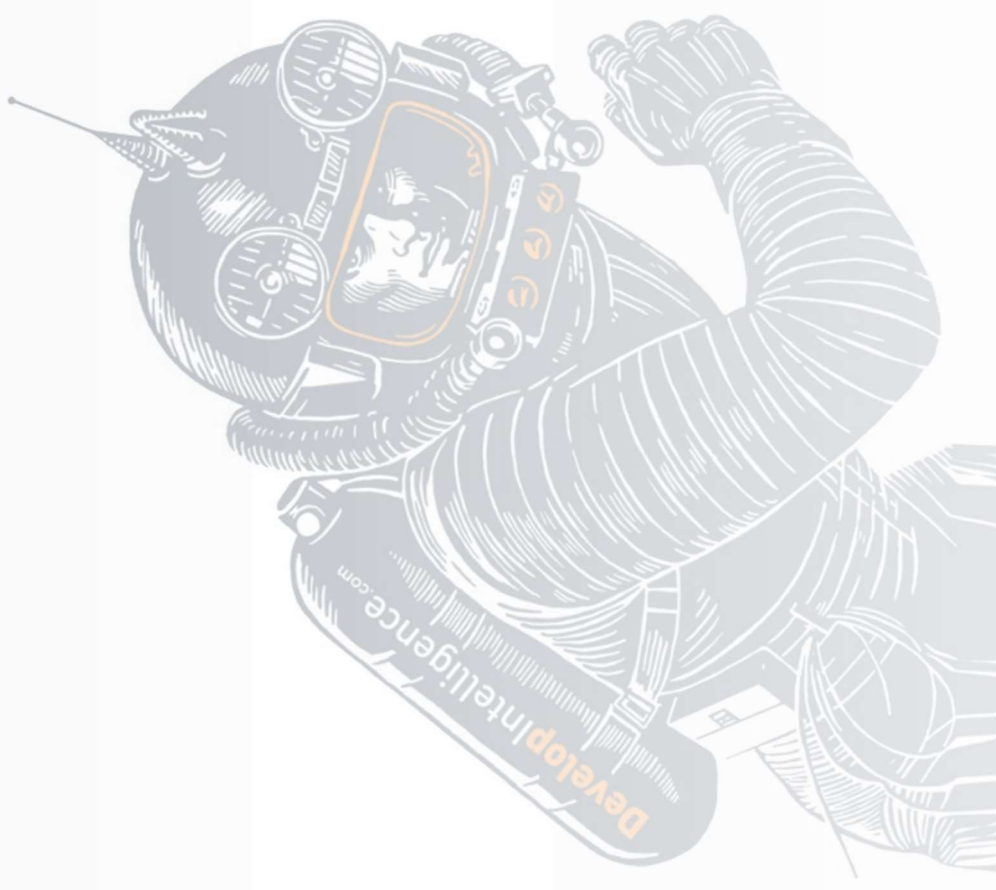
Security Theater

- Lots of security practices don't help
 - Password expiration
 - PIN complexity
- Do it anyway
- Tall poppy syndrome



Resources

- Sites
 - Schneier on Security
 - OWASP Cheat Sheet Series
 - Krebs on Security
- Books
 - Kevin Mitnick
 - The Gift of Fear
 - Facing Violence





Review

- Discuss the merits of security theater
- List 2 of the basicest rules

