embla Software Innovation

2017

TechDeep

Tech | Science | A Lot More!

www.embla.asia

CONTENTS

Editorial

"Augmented Reality is an emerging technology which superimposes a computer-generated image on a user's view of the real world, thus providing a composite view. We see a drastic increase in the popularity of Augmented Reality where 'Trends make ideas come to fruition' - and what better way to stay ahead of the game than to predict these trends yourself."

The phrase Augmented Reality (AR) which was coined in early 2000s means "add something" to the reality as we see it. Unlike virtual reality, which requires you to inhabit an entirely virtual environment, augmented reality uses your existing natural environment and simply overlays virtual information with, graphics, sounds, and touch feedback to give more user viable and believable experience. As both virtual and real world harmoniously co-exist, users of augmented reality experience a new and improved world where virtual information is used as a tool to provide assistance in everyday activities.

With the technology moving towards Artificial Intelligence and a "connected world" of internet of things, augmented reality will play a major role in developing a more holistic experience, not only for the humans but also for all connected applications. For example, a self-driving car in the future is most likely to use an augmented reality based traffic lights or the road signs which will not only reduce the cost and the clutter on the roads, but also make the experience smooth and futuristic.

The first edition of Embla TechDeep Magazine is focused more on emerging technologies such as Augmented Reality, and it is ultimate guide for those who wish to start their career in a similar field. In addition to this, the magazine also includes technical articles written by our developers, such as; XSS, Progressive Web Apps, SQL Best Practices and so much more.

We greatly appreciate the support we had from all our colleagues at Embla in order to make this magazine a success despite of their busy schedules. All articles in TechDeep is written by our developers. As the first step, this magazine will be hosted on our Website in PDF format. If you wish to share your feedback, we welcome your thoughts for upcoming editions of TechDeep magazine via our email magazine@embla.asia.

Chathuranga Bandara

Sugandika Jayasinghe

First day at work is always a challenge

Big changes in life make most people uncomfortable. In 2008, I left what in many reports have defined as “The best country in the world” to arrive in Sri Lanka, which has been fighting the civil war for over 25 years.

Coming from a society with plans made decades into the future, to a culture where most is wondering where the food for the next 14 days will come from – is a CHALLENGE. Some look at me and use the term idiot or totally crazy. I look upon it as a never-ending REWARD.

What can be more rewarding than greeting close to 60 smiling colleagues in the morning? What can be better than looking back on 8 years of building a high-tech IT-company? What can even come close to be married to a local Sri Lankan Princess? The sun never goes down. This is my never-ending reward.

When achieving such greatness, it is easy to forget that we are not alone. OUR company is a shared achievement by dedicated, hardworking individuals with a good mix of beliefs and inheritance. When interacting with customers from Scandinavia and Europe on a daily basis, it is all about people and their culture.

Our “ Extended Office” trademark have brought businesses, people and cultures together in ways we never thought would be possible 8 years ago. Together, we are not only building a company – we are building better lives – we are building a healthier nation – a new PARADISE.

Every day I wake up, I look forward to join forces with my “extended family” in office and all around the world. The first 8 years in Sri Lanka has been the ultimate reward and I look forward to spending each day with my extended family, as long as I live….

Stein Inge Haaland

Power to Empower

Empowerment and growth are not just integrated, but inseparable. This belief is at the heart of Embla's most important value – 'We empower innovation and growth'.

Empowerment puts more possibilities into play, and inclusion – going beyond prevailing, predominant or traditional perspectives – makes tapping into those possibilities more likely. We anchor entrusting our values and we embed it in our business, to improve our ability to connect and respond in a changing world.

It has always been the right thing and the smart thing to do. We believe, that it is a powerful recipe of success and, in turn will give an individual a sense of accomplishment and authority to make a difference. By coming together and sharing our unique perspectives, we grow as individuals as well as an organization that leads to new insights and innovation.

Unified by our shared vision and values which are enriched by individual authority and delegation, Embla and its people are stronger than ever. This helps us attract and retain the best, while helping us serve better and meet the needs of our clients.

We're proud of what we've achieved in our journey and we're motivated by a strong sense of purpose for what is still to come. We believe, that authorization and enablement strengthens us and we are committed long-term to progress in our company and the communities we serve.

Simply having empowerment is interesting, but doing something with it is powerful.

Chandimal Wickramaratne

Fishtalk Parser

Dananji Liyanage

In 2013, almost 20% of the protein need of the world population was satisfied with seafood. And it was recorded that nearly 50% of the need for seafood in 2014 was fulfilled with farmed fish [1]. According to the World Bank, by 2030 nearly two-third of the seafood will be farm-raised [2].
Therefore it is essential to increase farm-raised seafood production while preserving the environment. Among the farm-raised fish, Salmon is one of the most popular species, and Norway is the largest Salmon farming country in the world.

Among all the livestock industries, aquaculture yields the highest Feed Conversion Ratio (FCR), in other words output to input ratio. Therefore the most expensive input in aquaculture is the fish-feed, which accounts approximately up to 60% of the total cost of production. Hence, the wastage of fish-feed is directly proportional to the loss incurred.

In order to determine the quantity of fish-feed, a farmer should have a broad knowledge of nutritional facts and pellet size (diameter of feed) - fish feed quality - of the respective feed. This is somewhat a difficult task and lack of knowledge in the fish-feed quality results in the following issues:

1.Overfeeding

2.Environmental pollution

3.Death of both wild and farmed fish

4.Extinction of wild fish species

5.High sedimentation in sea bed (chemical imbalance in benthic environments)

FishTalkParser is developed to solve and control these scenarios. An overview of the product is shown in Figure 1.

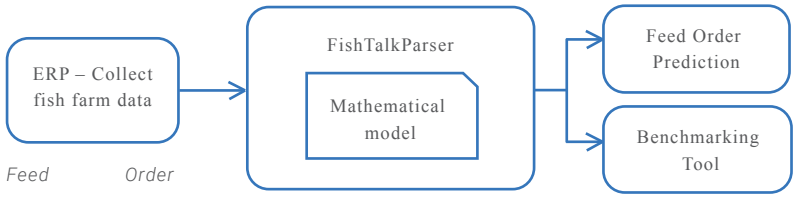


Figure 1: Product overview

Prediction

This module gives a fish farmer the ability to manage and control the amount of fish-feed dumped into the tanks.

his calculation uses a highly accurate mathematical model developed based on the empirical data, reflecting the growth of the fish. The module gives the quantity of the different quality fish-feed needed to feed the fish, which is illustrated in Figure 2.

Benchmarking Tool

This module contains two sets of graphs. The first one visualizes four standard statistics to measure and compare the status of the fish farms.

1. SPI – Sediment Profiling Imagery
2. SUR – Survival Rate
3. RGI – Relative Growth Index
4. FCR – Feed Conversion Ratio

These statistics help the fish-farmers to compare the status of their farm with other fish farms. This creates conversation within the community, which enables knowledge sharing. Through this program, common problems are solved within a shorter time frame, leading to higher yield.

The other set of graphs indicate time series data for temperature, growth, and feed consumption. Allowing fish-farmers to visualize the growth of their own farms enables them to predict the yield from the farm at the end of fish-cycle, labor and fish-feed requirements in the future.
Benchmarking tool as a whole, gives both the fish-farmer and the feed producer the ability to visualize the behavior of the fish farms throughout its life cycle.

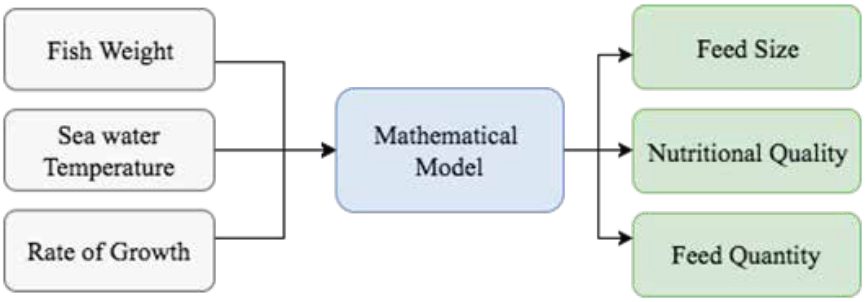


Figure 1: Product overview

Technology & Quality

The uniqueness of this product in respect to other similar products in the market depend on the efficiency of the program. The distinguish factor is the use of Haskell [3], an open source pure functional programming language. Performance benefit is mainly due to lazy evaluation in Haskell language. To give some insight about the run-time statistics of the program, Figure 3 shows processing of an input file of size 2.26 GB.

FishTalkParser uses a SQLite backend database, which contains all the settings needed for the calculations. Use of SQLite in this has made the configuration of the product in any environment easier and fast.

This product won the Bronze award under the Sustainability and Green IT category in National Best Quality ICT Awards, Sri Lanka 2016.

```
2,331,921,103,328 bytes allocated in the heap
 129,674,561,008 bytes copied during GC
   1,934,513,712 bytes maximum residency (2677 sample(s))
      38,932,808 bytes maximum slop
            3808 MB total memory in use (0 MB lost due to fragmentation)

                                     Tot time (elapsed)  Avg pause  Max pause
  Gen  0      4501413 colls,     0 par   159.031s  162.318s     0.0000s    0.0254s
  Gen  1         2677 colls,     0 par   251.609s  254.070s     0.0949s    2.6965s

  INIT    time    0.000s  (  0.009s elapsed)
  MUT     time  836.375s  (1040.135s elapsed)
  GC      time  410.641s  (416.388s elapsed)
  RP      time    0.000s  (  0.000s elapsed)
  PROF    time    0.000s  (  0.000s elapsed)
  EXIT    time    0.031s  (  0.047s elapsed)
  Total   time  1247.047s  (1456.579s elapsed)

  %GC     time      32.9%  (28.6% elapsed)

  Alloc rate    2,788,128,654 bytes per MUT second

  Productivity  67.1% of total user, 57.4% of total elapsed
```

References

1. https://www.msc.org/healthy-oceans/the-oceans-today/fish-as-food
2. http://wtop.com/food/2015/06/what-you-need-to-know-about-farm-raised-vs-wild-caught-fish/
3. https://www.haskell.org/

Progressive Web Apps

Dulitha Ruvin Daluwatta

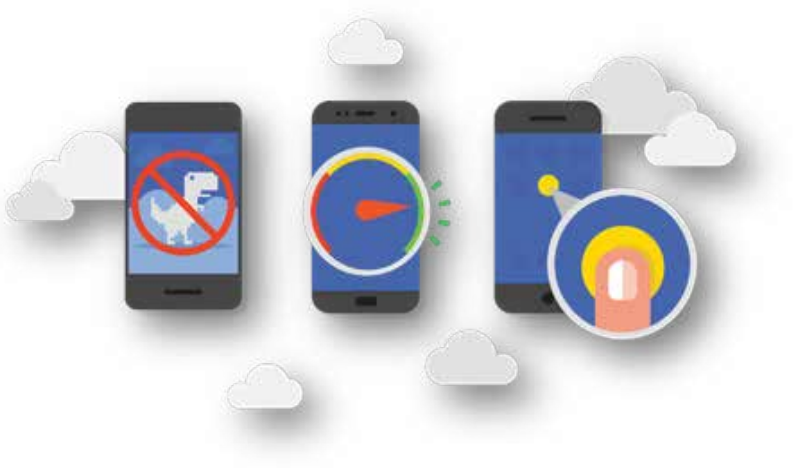
What are Progressive Web Apps?

Somehow, you may have heard that Progressive Web Apps are the future of all mankind which will bring world peace, save the earth, end world hunger and all good things we are waiting for. Maybe, it will do just all that and more in the future, but in the meantime let's see what they are for the time being, shall we?

Native mobile applications do functions such as send push notifications, offline working capability, loading from the home screen of the device and most of all, the look and feel like an app (as Google and Apple app overlords have imagined them). But since the beginning of the mobile friendly web apps, they weren't able to deliver these features we expect from a native app. However, Progressive Web Apps are here to fix that with all its new fancy web APIs, new design and design concepts.

"A Progressive Web App uses modern web capabilities to deliver an app-like user experience."

developers.google.com



Progressive Web Apps bring native application features to the mobile browser, that uses standards-based technologies and run in a secure container accessible to anyone on the web.

Main Characteristics of Progressive Web Apps

• Reliable?

Loads instantly without never showing the usual "no internet connectivity" messages even if there is no connectivity for the device. When launched from the user's home screen, service workers enable a Progressive Web App to load instantly, regardless of the network state.

• Fast

Super responsive to the users command without any lagging like in the usual web based applications (browser based).

• Engaging

Progressive web apps are intended to be installable and to live on user's home screen, not being constrained by an app store, offers an immersive full screen experience with the help of the web app manifest file and can be even re-engage users with push notifications. [1]

Some of the Core Tenants of Progressive Web Apps

• Service Workers

Service Workers are an incredibly powerful, and equally as confusing technology behind a Progressive Web App. They power push notifications, offline functionality, content caching, background content updating and a whole lot more. At a high level, a Service Worker is a worker script that works behind the scenes, independent of your app, and runs in response to events like network requests, push notifications, connectivity changes, and more. [2]

• App Shell

The App Shell model is a simple design concept whereby the initial load of a mobile web app provides a basic shell of an app UI, and the content for the app is loaded after. App Shell isn't a Web API or a framework, but rather a design approach that developers can choose to adhere, which is enhanced by the caching abilities of service workers.

You might find that it's a pretty straightforward, obvious approach, made more dramatic by a buzzword like everything with new tech these days. [2]

• Install ability and App Manifest

This is one of the major features of Progressive Web Apps. Historically, it was impossible to install mobile web apps like a normal app to the home screen. Therefore, the user had to pin a mobile website to the home screen on Android or IOS. But it was not even close to the expected native app experience.

Things are changing rapidly. Chrome on Android has added support for installing web apps to the home screen with a native install banner, just like the native app banners we're used to. [2]

Image Courtesy

1. http://umarani.com/progressive-web-apps-are-coming
2. https://developers.google.com/web/progressive-web-apps/

References

1. http://blog.ionic.io/what-is-a-progressive-web-app/
2. https://developers.google.com/web/progressive-web-apps/

$2 Arduino Projects

Malith Thilakarathne

Do we really need the $20 Arduino board to run our projects?

I came across this question as I was doing a project that needed to be placed inside a bulb holder. So, I researched through the internet and found some interesting ways to shrink an ordinary Arduino project. This article explains one of the ways I practically put into effect.

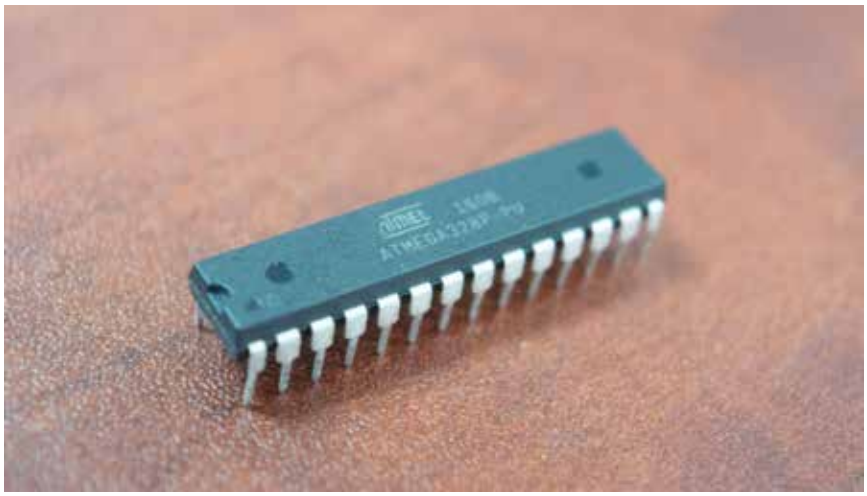
What we are going to do?

Program an ATmega328 chip using an Arduino Uno board and replace the entire Arduino board with that ATmega328 microcontroller and **few more extra parts.**

ATmega328 chip is the microcontroller chip that you see on the Arduino Uno board.

What are those "extra parts"?

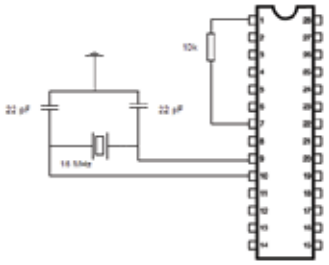
- one 16 MHz crystal,
- one 10k resistor
- two 22 pF capacitors (ceramic).



How do we do it?

First develop and test your project using the Arduino board as usual. Then take the ATmega chip out of the Arduino board carefully.

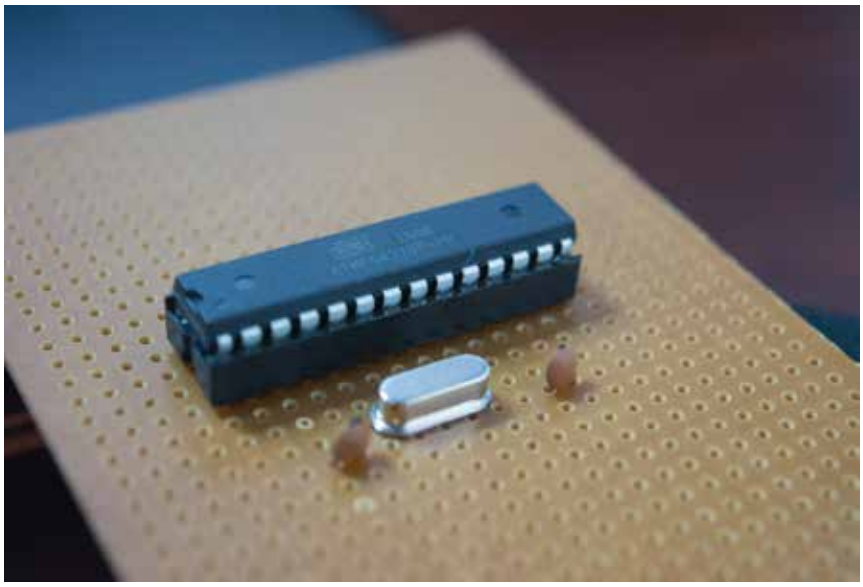
Well done! Now you have your working code programmed in to the ATmega chip. Connect the "extra parts" as follow.



The 10k resistor acts as a pull-up resistor to ensure that the reset pin stays at either a high or low state. (prevents floating)

Then complete the rest of your project using the following pin mapping. Remember that you need to power the ATmega chip (safe maximum 5.5V) using the VCC pin and GND.

If you can use a 28 pin DIP socket to add the ATmega328, you can re-program it easily after taking back to the Arduino board. And vice versa.



Now you don't have to disassemble your loving projects anymore! you can keep them forever by spending only 2 dollars. You can even go to production.

References

https://medium.com/@malith/2-arduino-projects-a7c552b5e7e

Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Ref – https://www.arduino.cc/en/Hacking/PinMapping168

Simple Rules for a Simple Design

Manik Navarathna

Keeping the code simple and clean is not an easy task. However, in 1990, Kent Beck came up with four simple rules in making a design simple whilst leaving the code cleaner in the process. The four rules that make a design simple are as follows in the order of its importance:

- Runs and passes all the tests
- Contains no duplication
- Expresses the intention
- Minimizes the number of elements

Runs and passes all the tests

Even though tests are a part of programming itself, they act as a marker which indicates a system works as expected. Unit tests are helpful in identifying flows in design and violations of basic principles such as SRP (Single Responsibility Principle). When the tests get harder to implement and integrate, it implies a violation of basic practices and clean code

Hence, the developers have to clean the code, apply basic design principles and introduce practices like DIP (Dependency Injection Principle) whilst focusing on loose coupling and higher cohesion.

Additionally, tests are helpful in keeping the code clean after implementation. Since the logic is pro-tected by the tests, developers can go ahead and refactor the code.

No duplication

DRY (Don't Repeat Yourself) is a powerful advice in protecting a code from the most dangerous rivalry in clean code, the duplication. A duplicate code not only add unwanted elements but also makes the system more complex and a lot harder to maintain as it minimizes the lack of clarity in the code. In oth-er words, duplicate code makes the code messier.
Two code pieces that look alike could be duplicates. However, duplicates can occur in many other forms.

For instance, assume that following are two method contracts in a collection class.

```
int size() {}

Boolean isEmpty(){}
```

In the above mentioned case, we can implement both methods in two different ways. Or else, we can eliminate the duplication by calling the 'size' () method in the 'is Empty' () method.

Duplication is a clear implication of violating a SRP. Techniques like Template Method Pattern can be used to reduce high-level duplications.

Expresses the intention

This simply means that a code should be understandable by the readers. In other words, a reader should be able to understand the purpose of the code when reading it. This is very important at the maintenance phase of a system because other developers will have to go through and understand the intention of the code pieces when investigating defects or adding new features into the system. Ex-pressive clear code will save a huge amount of time and cost (cost of delay).

Usage of meaningful names for methods, classes, variables and components will be helpful in making a code more expressive. Large functions can be broken down into smaller functions. In addition, we can use the standard nomenclature as well. For an instance, design patterns can use the names in the proper class as “Factory”, “Builder” etc.

Minimizes the number of elements

This rule implies that, whichever code that does not adhere to the previous rules should be removed. In addition, this rule might help in identifying too much separation of classes and methods. Some de-velopers tend to use too much interfaces, even where it’s not necessary. And some tend to make too much separation between behavior and fields in classes. This rule helps identify these situations. However, it should be noted that this rule has the least priority among the four rules.All the developers at some point write bad code and release. This is due to the deadlines or lack of knowledge. A good and clean code is not easy to implement. However, experts with many years of experience have found better practices and documented them for the ease of next generations.

These principles and practices not only help the developers keep the code elegant, but as an outcome, help reduce the additional costs caused by bad code.

References

Clean code – A Handbook of Agile Software Craftsmanship by Robert C. Martin

1. http://martinfowler.com/bliki/BeckDesignRules.html - Martin Fowler
2. http://blog.jbrains.ca/permalink/the-four-elements-of-simple-design

Cross-site Scripting Attacks:
Misunderstood and Dangerous

Nirmal Lanka

With Cross-site scripting (abbreviated XSS), most popular security mechanisms including firewalls and encryption become completely irrelevant. Also, due to the fact that the hijacking a session of a customer or an administrator of a web application —one of the results of an XSS attack— can have massive consequences on the business value of a service. Such a vulnerability can be much more devastating than most other attacks.

How misunderstood is it?

In a large number of situations, cross-site scripting is compared to SQL-injection due to similarities in their practice of injecting malicious code into legitimate and trusted code. However, this is misguiding and demeaning to both attack categories. They are two quite independent beasts that work in very different contexts and arise from different motivations.Unlike SQL-injection, cross-site scripting doesn't inject malicious instructions into database-access processes at the hands of a malicious user, but rather work by manipulating an innocent end-user into executing code that will be seen by the server as legitimate user activities. While the prevention of SQL-injection can be (arguably) easily achieved to a satisfying degree as consequence of techniques like intermediate query frameworks and ORM (especially in the .NET world), cross-site scripting (or XSS in street-talk) is not usually considered a threat due to the lack of understanding of its severity and state of reliable resolutions. Even the name "cross-site scripting" itself confuses most parties. While this name made sense with the scenarios in which it was discovered, it has long surpassed the security of same-origin policy , with the examples of stored XSS and second-order XSS .

Introduction to Cross-site scripting

The original idea of an XSS attack was that a website could cross a boundary and run arbitrary code inside another page; spy over sensitive data in forms, re-write page content and so on; hence the name cross-site scripting. And this is why same-origin policy was introduced as a prevention form. But hackers perceived this only as a challenge and invented novel ways to circumvent the security by adapting the same concepts. Therefore, while the name "cross-site scripting" made sense with the scenarios in which it was discovered, cross-site scripting has long surpassed the security of such crude security mechanisms , with the examples of stored XSS and second-order XSS. In simple terms, cross-site scripting as we know today, is an umbrella phrase covering a broad range of attacks that ultimately execute malicious JavaScript in the context of a website.

Persistent XSS (or stored XSS) stores strings containing the malicious code inside the web application's database to be later inadvertently executed (on viewing) by a victim . These can be posts or comments in a forum, a user-to-user message or a HTML-customized page of a social profile.

Reflected XSS (or non-persistent XSS) embeds such malicious strings inside a request sent by the user (under manipulation) to the application , for example; as an AJAX GET request

In addition to that, a **DOM-based XSS** attack is a situation where the whole attack takes place inside browser, without depending on the back-end to take any part . Such attacks rely on the existing, legitimate JavaScript of a page (even frameworks like AngularJS) to help execute the malicious instructions and generally originate from a URL.

Outcomes of an XSS attack

As in many vulnerabilities, the application of a well-planned cross-site would usually be simply a stepping stone in a much larger scheme .With XSS, the harvest of user credentials —or more commonly user sessions leading to exactly that— becomes trivial.
Hijacking the victim's cookies containing the session ID can be as simple as calling document.cookie and sending it as a simple GET request to the attacker's server.

Phishing

Attacks where unsuspecting users are manipulated into clicking seemingly innocent links inside social networking sites (including but not limited to Facebook), simply viewing a certain social forum post or falling for a dynamically created page that is disguised as a privacy or security reminder are quite common. Such phishing attacks have the potential to be extremely difficult to separate from legitimate content and are not identifiable through simple vigilance as in the case of traditional phishing attacks where a user is persuaded into visiting a fake login screen in an attacker-controlled domain .

Keylogging

board event-listener through addEventListener(..) is just a matter of coming up with a clever pattern for breaking through the encoding or sandboxing of any client-side framework. The astounding and fearful nature of being able to track all keystrokes of a user in any web application requires no further explanation .

The potential scope of an attack

Prevention

Simplest and most common forms of attempts to prevent cross-site scripting is **encoding** and validation. Encoding is used for escaping the user input and helping the browser interprets such strings only as data and not code. **Validation** is used for filtering user input in cases where it is required to be executed, by helping the browser strip or block malicious instructions .

Such careful handling of user input must to be done differently depending on the context. In other words, where in a page user input is inserted. This can be performed either at the reception of input or prior to the insertion of that input into a webpage.Sanitization or handling of input can be done on the side of the user-agent (browser) or on the side of the back-end, depending on the circumstance. The same input may be required to be inserted or processed at different places and under different contexts.Such security measures need to be taken without limiting the user-experience and functionality.

Validation

Blacklisting and whitelisting whitelisting are two methods used for validating, i.e. sanitizing or rejecting input. Both have competing advantages as well as drawbacks. But generally, blacklisting, which is a method of preventing known patterns is deemed inferior to whitelisting, which only allows extremely selective input patterns .
Still, secure input handling alone may not be a long-term solution in many cases, as oversight in a single place can render security in all other places worthless.

Content Security Policy

As a firm and reliable solution for such situations, CSP introduces policies for the browser to prevent ground for such vulnerabilities.

While parameters of CSP are well customizable , three main categories can be identified:

- **No untrusted sources** – Content can be loaded only from a set of clearly-defined external sources.
- **No inline resources** – No inline JavaScript and CSS will be executable.
- **No eva**l – The eval() function in the JavaScript API will be blocked.

Conclusion

Admitting the possibility of a problem is the key to finding a resolution for it. With familiarity and foresight, most unfortunate circumstances can be expected and avoided. Otherwise, every speck of effort in securing a product can be washed-away due to simple oversight.

Cross-site scripting is a very critical scenario for the existence of which every web-application developer should be constantly concerned about and implement the best of countermeasures. Lack of clear understanding of the severity of the threat it poses and its attack vectors lead to the loss of motivation and knowledge required in developers for identifying related critical vulnerabilities in their applications, ultimately putting them on a false sense of security.

References

"Confused about XSS vs Injection attacks?" – https://community.rapid7.com/community/nexpose/blog/2013/06/20/xss-vs-injection

"Security Considerations (Entity Framework)" – https://msdn.microsoft.com/en-us/library/cc716760(v=vs.110).aspx

"Fixing SQL Injection: ORM is not enough" – https://snyk.io/blog/sql-injection-orm-vulnerabilities/

"Bypassing Same Origin Policy (SOP)" – http://resources.infosecinstitute.com/bypassing-same-origin-policy-sop/#gref

"Web Application Exploits and Defenses (Part 2)" – https://google-gruyere.appspot.com/part2

"Bypassing Same Origin Policy (SOP)" – http://resources.infosecinstitute.com/bypassing-same-origin-policy-sop/#gref

"Web Application Exploits and Defenses (Part 2)" – https://google-gruyere.appspot.com/part2

"Part One: Overview" – https://excess-xss.com/

"Reflected XSS attack exploitation on victim's domain" – figure – https://www.researchgate.net/figure/279869021_fig3_FIGURE-3-Reflected-XSS-attack-exploitation-on-victim's-domain

"admin.google.com Reflected Cross-Site Scripting (XSS)" – https://buer.haus/2015/01/21/admin-google-com-reflected-cross-site-scripting-xss/

"DOM Based XSS" – https://www.owasp.org/index.php/DOM_Based_XSS

"Eliminating the myths of XSS attacks" – http://qnrq.se/eliminating-the-myths-of-xss-attacks/

"Session hijacking attack" – https://www.owasp.org/index.php/Session_hijacking_attack

"Applying XSS to Phishing Attacks" – paper – http://www.xssed.com/article/5/Paper_Applying_XSS_to_Phishing_Attacks/

"xss-keylogger" – https://github.com/hadynz/xss-keylogger

See [xvi]

See [xv]

"XSS myths: input validation is not enough!" – http://blog.7-a.org/2011/04/xss-myths-input-validation-is-not.html

"Content Security Policy Level 3" – W3C Working Draft, 13 September 2016 – https://www.w3.org/TR/CSP/

"Content Security Policy" – https://developers.google.com/web/fundamentals/security/csp/

Dream Big

Sharon Kern

As humans we all have dreams in life, be it personal or professional. Dreams differ from one person to another. Dreaming involves holding tight to a vision for a better life. However, getting there might be difficult, having to deal with setbacks and failures along the way

In order to make dreams a reality, it needs action. In simple terms, it only needs the will power and the belief within. Dreaming big motivates and gives direction. A dream is a goal that a person will work towards. A person who dreams have a purpose in life and strives to make it a reality. Dreamers are believers, and they are busy working on their dreams.

Inspiration

Having a dream will boost self-confidence when you dream the impossible. If someone says "I will start my own business in five years", you will know that this person is a confident person.

If you do not dream big, you are limiting yourself. Not all dreams will turn in to reality. Yet, trying to achieve it and making the ground work will be an achievement.

Dreams make you stronger

When there are challenges to confront failures, always be a part of it and that makes a person stronger to do better next time. If there are no failures, chances of trying would be at a minimal or zero. It takes a lot of courage to believe that a person can do something impossible, until it's been done.

"If you aim for the sun you can fall on the stars"

There is a famous saying that "If you aim for the sun, you can fall on the stars". In most instances, humans always aim low as they are either happy in their own comfort zone or not willing to make an extra effort to strive hard. Aiming high has its advantages. For example; you would at least achieve in between.

This simply means that the effort put in will not be wasted. Dreams have to be big in order to reach the expected level. And one must believe in a dream to make it a reality.

> "If your dreams don't scare you they're not big enough"
>
> *Anonymous*

Augmented Reality Application Development

Sith Indunil Withana

Augmented Reality (AR) is the blending of virtual reality and real life, as developers can create images within applications that blend in with contents in the real world. With AR, users are able to interact with virtual contents in the real world, and are able to distinguish between the two. [1]

Figure 1 - Virtual Reality Mahindagamanaya

As figure 1 above, virtual reality has approached a new level where people use it to demonstrate significant historical moments in our life.

Also, there are products such as Google glasses and Microsoft HoloLens which uses virtual reality and augmented reality in order to perform day to day activities such as reading mails and generating weather reports.

Geroimenko, 2012, defined augmented reality as;

"A real-time direct or indirect view of a physical real-world environment that has been enhanced/augmented by adding virtual computer generated information to it" [2]

Features of augmented reality as per Geroimenko, 2012:

- It combines the real world with computer graphics.
- It provides interaction with the objects in real time
- It tracks objects in real time
- It provides recognition of images or objects
- It provides real-time context or data

Basically augmented reality has two main components,

- Computer generated images, videos or shapes
- Real world objects

AR application development has two main approaches,

- Marker less Augmented Reality
- Marker based Augmented Reality

Marker less Augmented Reality Approach

In order to perform an augmented reality task on a real environment, the system must use an artificial or real world marker which can be either an image or an object. Therefore, this method will use a part of a real scene as a marker. (Geroimenko, 2012)

Advantages and Disadvantages of Marker less approach

Advantages	Disadvantages
• No need of a fixed marker (Geroimenko, 2012) • Uses the real object as the marker	• Needs more complex algorithms to identify object. (Geroimenko, 2012) • Needs to program for angles and corners of the object in order to identify. (Geroimenko, 2012) • Needs to consider lighting conditions. • Needs to identify colors in the object vclearly.

Table 1- Advantages and Disadvantages of Marker less approach

Marker based Augmented Reality Approach

Markers can be divided in to two parts,
• Printed Markers
• Natural Markers

In general, these markers are used as a point of reference which defines the position, orientation and scale of the Augmented Reality object and, applications use greyscale markers such as QR codes in order to identify the marker and position. When looking at the history, it can be seen that markers have evolved from Barcode Readers to Real life markers.

Advantages and Disadvantages of Marker Based Augmented Reality Approach

Marker Type	Advantages	Disadvantages
• Printed Marker	• Greyscale images, so no need to focus on color changes in the image • Easy to create the image (Geroimenko, 2012) • Lots of third party apps to create custom markers (QR code generator) (Geroi menko, 2012) • Simple patterns	• Only can be used on flat surfaces (Geroimenko, 2012)
• Natural Markers	• No need of any external code to be printed or made (Geroimenko, 2012) • Natural and no modification needed	• Needs complex algorithms to identify the markers • Needs to identify different patterns and features of the natural markers. •Different natural states such as lighting has an impact on identifying the marker (Geroimenko, 2012) • Difficult to set bound and scope of the AR content

Table 2 - Advantages and Disadvantages of Marker less approach

Normally, developers use the marker based approach due to the simplicity and less development time. We'll discuss more about the AR application development in the upcoming versions of **Embla Magazine.**

References:

1"Fixing SQL Injection: ORM is not enough" – https://snyk.io/blog/sql-injection-orm-vulnerabilities/

2 Geroimenko, V., 2012. Augmented Reality Technology and Art: The Analysis and Visualization of Evolving Conceptual Models, Montpellier: International Conference on Information Visualisation.

SQL Best Practices

Tharsan Sivakumar

A best practice is a technique or methodology which through experience and research, has proven to reliably lead to a better result. Throughout the software industry, several best practices are widely followed and every programming language has best and worst practices, and SQL is also no exception. This article discusses only 3 simple best practices for the routine mistakes that we do. The color codes for the code snippets can be interpreted as below.

Routine practice

Best practice

Practice #1 – Do avoid use of **SELECT** * in SQL queries

Instead of writing the queries with * operator, do practice writing queries with required column names. This technique results in reduced disk I/O and better performance. When the query is written with * operator, that operator has to be resolved as what does that operator mean with the table's meta data.

The performance will degrade as the server engine has to fire an additional background query to resolve that operator.

```
SELECT * FROM Product
```

Figure -1

```
SELECT Id, Name, ProductNo,
StandardCost  FROM Product
```

Figure -2

Practice #2 - Use JOINs for better performance than sub queries

In most cases, JOINs are faster than sub-queries and it is very rare for a sub-query to be faster. The RDBMS can create an execution plan that is better for your query and can predict what data should be loaded to be processed and save time, unlike the sub-query where it will run all the queries and load all their data to do the processing. The good thing in sub-queries is that they are more readable than JOINs, but when it comes to performance, JOINS are better.

```
SELECT SalesOrderId,
ProductId FROM
SalesOrderDetail
WHERE ProductId IN (SELECT
ProductId FROM Product
WHERE ProductSubCategoryId
IN (SELECT
ProductSubCategoryId FROM
ProductSubCategory WHERE
Name = 'MOUNTAIN BIKES'))
```

Figure - 3

```
SELECT s.SaledOrderId,
p.ProductId
FROM SalesOrderDetail AS s
INNER JOIN Product p
ON s.ProductId = p.ProductId
INNER JOIN
ProductSubCategory ps
ON ps.ProductSubCategoryId =
p.ProductSubCategoryId
WHERE ps.Name = 'Mountain
Bikes'
```

Figure - 4

Practice #3 – Do avoid functions in WHERE clause

```
WHERE DATEDIFF(yyyy, Person.-
DateOfBirth, GETDATE()) > 21

WHERE SUBSTRING(Person.Last-
Name, 5) = ''Beck'
```

Figure - 5

```
WHERE Person.DateOfBirth >
DATEDIFF(yyyy, -21,
GETDATE())

WHERE Person.LastName LIKE
'Beck%'
```

Figure - 6

When we wrap a function around an indexed column, SQL Server must compute the value of the function for each row in the table.

As it's going to be performed in run time, it will cost a lot. Furthermore, when we wrap the column with the function, it may ignore the index as well. When the query is written as shown in Fig -5 (function applied to left hand side), it has to calculate the value for each row in run time, where as if the same query is written as in Fig -6, it will evaluate the values in each row with one time calculated value on the right hand side. If you still want to wrap the column with the function, then create functional index column, so that the server will generate an index and keep it ready before hand; and the cost may be less.

Not only these, there are so many best practices we can follow in our day today development. Some of them are given below.

- Keep primary key of lesser chars or integer. It is easier to process small width keys.
- Use normalized tables in the database. Small multiple tables are usually better than one large table.
- Store image paths or URLs in database instead of images. It has less overhead.
- Use proper database types for the fields. If 'StartDate' is database filed use datetime as datatypes instead of VARCHAR (20)
- Use LIKE clause properly. If you are looking for exact match use "=" instead.
- Write SQL keyword in capital letters for readability purpose.
- Use stored procedures. They are faster and help in maintainability as well security of the database.

Closing note

There's always room for refactoring in every line of code we write. Sometimes, it could take a year or more before our original code becomes a problem. No matter how good we really are, the written code could be improved for both performance and readability by applying best practices.

References:

"Confused about XSS vs Injection attacks?" – https://community.rapid7.com/community/nexpose/blog/2013/06/20/xss-vs-injection
"Confused about XSS vs Injection attacks?" – https://community.rapid7.com/community/nexpose/blog/2013/06/20/xss-vs-injection
"Confused about XSS vs Injection attacks?" – https://community.rapid7.com/community/nexpose/blog/2013/06/20/xss-vs-injection

“It was a superb experience!

Malith Thilakarathne

Everyone was friendly, and we were treated equally - no hierarchies, no dress codes!
The best thing was that we were given the opportunity to work on real projects, and not some dummy projects. On the very first day, we participated in a scrum meeting of a live project. We were given an equal place in the team, work and as individuals. We received a valuable experience of client interactions and got to perform all the demos in front of the client. Especially during the last few weeks, I was able to handle my project's client by myself in daily scrum meetings.

The internship program at Embla is well structured and it really worked! We had induction sessions and continuous performance reviews. With that, we were able to improve ourselves as well as apply and practice the concepts we learn at university, in real life scenarios.

Life at Embla was really interesting. I was able to partake in Football, Badminton with Embla fitness; organizing the Halloween party and many other fun-filled events with Embla recreations, Tech talks, Lightening talks with Embla Voices, IOT projects with Embla IOT and so much more! I highly recommend Embla as THE BEST place to peruse your internship.”

Udara Bibile

"Internship experience at Embla was excellent.

Technically speaking, I got a great exposure to development where I had the chance to work with client projects. I was taught not only to code, but to code efficiently following best practices. In a cultural way, Embla has a very friendly atmosphere with lots of extra-curricular activities to keep us entertained. In conclusion, I could say that it was a privilege to work for Embla and I was lucky to be a member of the 'Embla Family' "

“I had a great internship experience at Embla.

We got to work in client projects and, I was able to gain an in-depth understanding of client requirements and so much more! Embla taught us how to do coding and to do it in the best way, following the best practices. Embla has a very friendly environment and the entire staff was very helpful. Most importantly, they treated us as equals. We also got to participate in a lot of interesting events and enjoyed a lot. I was very fortunate to be a part of the ‘Embla Family‘.”

Madhuri De Abrew

New Year Celebrations

Angular Meetup

Treasure Hunt

Embla Anniversary

embla recreations

Aluth Avurudu 2017

Wesak Dansala 2017


embla
Acoustica
EPL 2K17
RIU
Annual Trip
Gaming
Tournament
Halloween
Christmas and
Year end Party


SK
95
BUTTMAN

CREDITS

Writers

Fishtalk Parser
Dananji Liyanage

Progressive Web Apps
Dulitha Ruvin Daluwatta

$2 Arduino Projects
Malith Thilakarathne

Simple rules for a simple design
Manik Navarathna

Cross-site Scripting Attacks
Nirmal Lanka

Dream Big
Sharon Kern

Augmented Reality
Sith Indunil Withana

SQL best practices
Tharsan Sivakumar

Reviewers

Stein Inge Haaland

Chandimal Wickramaratne

Sharon Kern

Chamara Sanjeewa

Graphic Designers

Sandun Ramasinghe

Chamara Weerasinghe

Sri Lanka
144/A - 3rd floor,
Attidiya Road,
Dehiwala-Mount Lavinia

Phone
+94 112 761 696

Email
contact@embla.asia

Norway
Sandbyveien 25,
3512 Hønefoss

Phone
+47 91 82 60 55

Email
contact@embla.no

embla Software Innovation